

(19) BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENTAMT

(12) Offenlegungsschrift
(10) DE 195 23 036 A 1

(51) Int. Cl. 6:
G 06 F 9/45
G 06 F 17/60

(30) Unionspriorität: (32) (33) (31)
29.06.94 JP 6-171925 20.01.95 JP 7-26186

(71) Anmelder:
Marukin Corp., Yamanashi, JP
(74) Vertreter:
Vonnemann, G., Dipl.-Ing. Dr.-Ing., Pat.-Anw., 20099
Hamburg

(72) Erfinder:
Matsuzuki, Tadao, Tokio/Tokyo, JP

(54) Automatischer Programmgenerator

(55) Automatischer Programmgenerator, bestehend aus
- einem Bausteinspeichermittel zur Speicherung von Pro-
grammbausteinen, die Quellprogramme in halb fertiggestell-
tem Zustand darstellen,
- einem Entwurfsdokumentspeichermittel zur Speicherung von Entwurfsdokumenten in halb fertiggestelltem Zustand,
die den Inhalt der Verarbeitung der Programmbausteine
beschreiben,
- einem Eingabemittel für individuelle Informationen, dessen
Aufgabe darin besteht, während der Erstellung von Anwen-
dungsprogrammen im Dialog individuelle Informationen, die
mit den unterschiedlichen Anwendungsprogrammen varie-
ren, zu empfangen,
- einem Mittel zur Fertigstellung von Programmen, welches
die Programmbausteine auf Grundlage der erhaltenen individuellen Information editiert und Quellcodes für die entspre-
chenden Anwendungsprogramme erzeugt,
- einem Mittel zur Fertigstellung von Entwurfsdokumenten,
welches die in halb fertiggestelltem Zustand vorliegenden Entwurfsdokumente auf Grundlage der erhaltenen individuellen Informationen editiert und die den jeweiligen Anwen-
dungsprogrammen entsprechenden Entwurfsdokumente fertigstellt und
- einem Ausgabemittel, das auf eine entsprechende Hand-
lung hin eine Liste der Quellcodes und/oder des Entwurfsdo-
kuments jedes vollendeten Anwendungsprogramms aus-
drückt.

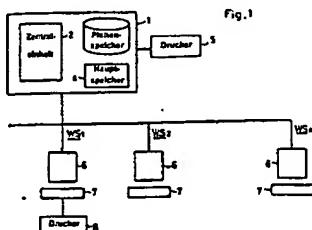
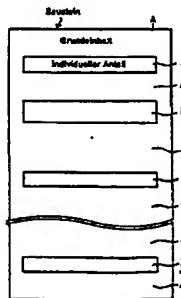


Fig. 2



DE 195 23 036 A 1

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

BUNDESDRUCKEREI 11.95 508 061/871

45/28

DE 195 23 036 A 1

Beschreibung

Die Erfindung betrifft ein automatisches Programmgenerierungssystem, das sämtliche Anwendungsprogramme auf einem gleichbleibenden Qualitätsniveau erstellen kann, zusammen mit einem standardisierten Entwurfsdokument, welches genau mit dem erstellten Programm übereinstimmt.

Die effiziente Nutzung eines Computers erfordert ein geeignetes Anwendungsprogramm, welches auf die Bedürfnisse des Anwenders zugeschnitten ist, bisher wurde die Erstellung dieser Anwendungsprogramme jedoch stark von den Eigenschaften des Programmierers beeinflußt.

Nach Fertigstellung des Anwendungsprogramms wird im allgemeinen dem Anwender der Maschinencode und das Handbuch für das Programm übergeben und das Anwendungsprogramm nach Angaben des Handbuchs betrieben. Im Falle einer Störung des normalen Programmablaufs wird Wartung angefordert, damit der normale Programmablauf erhalten bleibt. Nachdem das Programm fehlerfrei arbeitet, wird dem Anwender gelegentlich eine Aufstellung des Quellcodes sowie Unterlagen, in denen der Programmaufbau niedergelegt ist, übergeben, diese Unterlagen müssen jedoch von Hand erstellt werden, da die Fertigstellung sehr zeitaufwendig ist, erhält der Anwender nicht unbedingt vollständige Unterlagen.

In der Zwischenzeit mag der Anwender, um mit Neuerungen oder Änderungen seiner Dienstleistungen Schritt zu halten, den Wunsch haben den Programmablauf zu ändern, auch wenn das Anwendungsprogramm störungsfrei abläuft. Herkömmliche Hardware hat jedoch den Nachteil, daß ein Anwendungsprogramm nach seiner Fertigstellung nicht mehr einfach verändert werden kann.

Fehlt die Liste des Quellcodes des Programms oder weicht die Liste des Quellcodes, z. B. wegen der Programmfpflege, vom installierten Programm ab, erfordert daher selbst eine geringe Abwandlung im Programmablauf einen enormen Zeitaufwand, mit dem Ergebnis, daß ein neues Anwendungsprogramm von Grund auf wieder aufgebaut werden muß. Dies war in der Vergangenheit häufig der Fall.

Des weiteren besitzt jedes Programm, selbst wenn eine vollständige Aufstellung des Quellcodes vorhanden ist, eine ihm eigene Persönlichkeit, die von den Fähigkeiten und der Ausbildung des Programmierers geprägt wird, weshalb eine dritte Person den Programminhalt nicht einfach erfassen kann. Es kostet diese viel Zeit den Programminhalt zu verändern, in manchen Fällen hört das gesamte Programm wegen solcher Modifikationen auf störungsfrei abzulaufen.

Sollten verschiedene Unterlagen aus den Archiven zur Verfügung stehen, wird oft der Inhalt des aktuellen Programms nicht genau in den Unterlagen wiedergegeben, und selbst wenn der Inhalt genau beschrieben wurde, kann alles in unterschiedlichen Formaten abgefaßt sein. Auch dies erschwert es einer dritten Person den Programminhalt zu erfassen.

Da Anwendungsprogramme veränderten Bedürfnissen angepaßt werden müssen, sollte das ausgelieferte Programm die Möglichkeit einer späteren einfachen Überarbeitung und Erweiterung bieten. Genauer ausgedrückt sollten Anwendungsprogramme von gleichbleibender Qualität sein, nicht von der Persönlichkeit des Programmierers geprägt sein, und der Ablauf eines jeden an einen Anwender ausgelieferten Programms soll-

te so sein, daß er auch von jedem genau verstanden werden kann.

Unter Berücksichtigung der oben dargelegten Leitlinien ist es daher Aufgabe der vorliegenden Erfindung, 5 einen automatischen Programmgenerator zur Verfügung zu stellen, der automatisch Anwendungsprogramme gleichbleibender Qualität erstellen kann, die nicht von der Persönlichkeit des Programmierers beeinflußt werden und standardisierte Entwurfsdokumente bereitstellen, in denen der Inhalt des Aufbaus dargelegt ist.

Die Aufgabe wird dadurch gelöst, daß der erfindungsgemäße automatische Programmgenerator aufgebaut ist aus (1) einem Speichermittel für Programmbausteine, das Programmbausteine speichert, welche teilvollendete 15 Quellprogramme darstellen, (2) einem Speichermittel für Entwurfsdokumente, in welchem teilvollendete Entwurfsdokumente gespeichert sind, welche den Programmablauf der entsprechenden Programmbausteine beschreiben, (3) einem Mittel zur Eingabe individueller 20 Informationen, welches während des Programmablaufs jedes Anwendungsprogramms in Funktion tritt, um im Dialog individuelle Informationen, die zwischen den unterschiedlichen Programmen variieren, aufzunehmen, (4) einem Mittel zur Programmverbesserung, 25 das auf der Grundlage der eingegebenen individuellen Informationen die Programmbausteine editiert und Quellprogramme für die entsprechenden Anwendungsprogramme erstellt, (5) einem Mittel zur Verbesserung der Entwurfsdokumente, das auf Grundlage der erhaltenen individuellen Informationen das teilvollendete Entwurfsdokument editiert und ein Entwurfsdokument fertigstellt, das dem einzelnen Anwendungsprogramm entspricht, und (6) einem Mittel zur Ausgabe, mit dem die 30 Liste des Quellcodes und/oder des Entwurfsdokuments des fertig gestellten Anwendungsprogramms ausgegeben werden kann.

Das Programmbausteinspeichermittel speichert Programmbausteine in teilweise fertiggestelltem Zustand, so daß verschiedene Arten von Anwendungsprogrammen implementiert werden können. Diese Programmbausteine sind typischerweise eingeteilt in eine erste Gruppe von Programmen, die Daten in einer Übergabedatei speichern, einer zweiten Gruppe, welche die Daten in den Dateien suchen und am Arbeitsplatz anzeigen, einer dritten Gruppe von Programmen, die Daten in einer Stammdatei speichern, einer vierten Gruppe von Programmen, welche die Daten in den Dateien suchen und diese in einem Fenster am Arbeitsplatz anzeigen, einer fünften Gruppe von Programmen, welche den Inhalt der Dateien auf einem Formblatt ausdrucken, und einer sechsten Gruppe von Programmen, welche den Inhalt der Stammdateien und/oder der temporären Dateien entsprechend der Daten der Übergabedateien aktualisieren. In der vorliegenden Erfindung werden alle 40 Anwendungsprogramme unter Verwendung einer geeigneten Kombination der Programme aus der ersten bis sechsten Gruppe erstellt.

In der vorliegenden Erfindung ist zu jedem Programmbaustein ein Entwurfsdokument in teilweise fertiggestelltem Zustand abgespeichert. Auf Grundlage der im Dialog eingegebenen individuellen Information, wird das in teilweise fertiggestelltem Zustand vorliegende Entwurfsdokument editiert, auf diese Weise wird ein Entwurfsdokument erhalten, das genau mit dem einzelnen Anwendungsprogramm übereinstimmt. Da die Programmbausteine in sechs Hauptgruppen eingeordnet sind, können die Formate der Entwurfsdokumente und die Eingabefelder leicht standardisiert werden. Außer-

dem erleichtert es die Standardisierung der Entwurfsdokumente einer dritten Person den Inhalt jedes Dokuments zu erfassen.

Beschreibung der Figuren

Fig. 1 zeigt ein Blockdiagramm, in dem eine Ausführungsform des erfindungsgemäßen automatischen Programmgenerators dargestellt ist;

Fig. 2 zeigt in einem schematischen Diagramm den Aufbau des Programmbausteins;

Fig. 3 zeigt in einem Diagramm die Unterschiede zwischen den Entwurfsmustern 1–7;

Fig. 4 zeigt eine Darstellung eines typischen Einkaufsbelegs;

Fig. 5 zeigt eine Darstellung einer typischen Übergabedatei, in der die Daten des Einkaufsbelegs aus Fig. 4 eingetragen sind;

Fig. 6 zeigt ein Flußdiagramm, in welchem der Inhalt der Verarbeitung der Entwurfsmuster 1 und 2 dargestellt ist;

Fig. 7 zeigt eine Darstellung eines typischen Fensters der Bildschirmanzeige, um einen Wechsel des Betriebsmodus zu implementieren;

Fig. 8 zeigt eine Darstellung einer typischen Übergabedatei, in der die Daten des Einkaufsbelegs aus Fig. 4 eingetragen sind;

Fig. 9 zeigt ein Flußdiagramm, in dem die Bestandteile der Verarbeitung der Entwurfsmuster 3, 4 und 5 dargestellt sind;

Fig. 10 zeigt ein Diagramm, welches ein Beispiel für editierte Daten zeigt;

Fig. 11 zeigt eine typische Übergabedatei, in welcher editierte Daten eingetragen sind;

Fig. 12 zeigt ein Flußdiagramm, in dem die Bestandteile der Verarbeitung der Entwurfsmuster 6 und 7 dargestellt sind;

Fig. 13 zeigt ein Flußdiagramm, in dem die Bestandteile der Verarbeitung des Programmbausteins SK010 dargestellt sind;

Fig. 14 zeigt ein Flußdiagramm, in dem die Bestandteile der Verarbeitung des Programmbausteins SK010 dargestellt sind;

Fig. 15 zeigt genauer einige der Bestandteile der Verarbeitung des Programmbausteins SK010;

Fig. 16 zeigt ein Flußdiagramm, in dem die Bestandteile des Entwurfsmusters 11 dargestellt sind;

Fig. 17 zeigt ein Flußdiagramm, in dem die Bestandteile der Verarbeitung der Entwurfsmuster 12, 13 und 14 dargestellt sind;

Fig. 18 zeigt eine typische Stammdaten;

Fig. 19 zeigt ein Flußdiagramm, in dem die Bestandteile der Verarbeitung der Entwurfsmuster 15–29 dargestellt sind;

Fig. 20 zeigt eine Darstellung zur Erklärung der Kopierfunktion;

Fig. 21 zeigt ein Flußdiagramm, in dem die Bestandteile der Verarbeitung des Programmgerüsts WN010 dargestellt sind;

Fig. 22 zeigt eine Darstellung des Fensterdisplays;

Fig. 23 zeigt ein Diagramm, in dem die Bestandteile der Verarbeitung des Programmbausteins LT020 dargestellt sind;

Fig. 24 zeigt ein Diagramm, in dem die Bestandteile der Verarbeitung des Programmbausteins LT021 dargestellt sind;

Fig. 25 zeigt einen Ausdruck der Bestandteile einer Übergabedatei, der mit dem Programmbaustein LT021

erhalten wurde;

Fig. 26 zeigt ein Diagramm, in dem die Bestandteile der Verarbeitung des Programmbausteins LT010 dargestellt sind;

Fig. 27 zeigt Diagramm, in dem die Bestandteile der Verarbeitung des Programmbausteins LT030 dargestellt sind;

Fig. 28 zeigt ein Diagramm, in dem die Bestandteile der Verarbeitung des Programmbausteins LT040 dargestellt sind;

Fig. 29 zeigt ein Diagramm, in dem die Bestandteile der Verarbeitung des Programmbausteins BU020 dargestellt sind (Entwurfsmuster 35);

Fig. 30 zeigt ein Diagramm, in dem die Bestandteile der Verarbeitung des Programmbausteins BU010 dargestellt sind;

Fig. 31 zeigt ein Diagramm, in dem der Erstellungs- vorgang für eine physikalische Datei dargestellt ist;

Fig. 34 zeigt den Erstellungsvorgang für die Bildschirmdarstellung;

Fig. 35 zeigt einen Teil des Entwurfsdokuments für die Bildschirmdarstellung;

Fig. 36 zeigt einen anderen Teil des Entwurfsdokuments für die Bildschirmdarstellung;

Fig. 37 zeigt die Liste des Quellcodes, die für die physikalische Datei relevant ist;

Fig. 38 zeigt ein Diagramm, in dem der Vorgang für einen groben Entwurf des Programms (unit information registration) dargestellt ist;

Fig. 39 zeigt ein Diagramm, in dem der Vorgang für den genauen Entwurf des Programms dargestellt ist;

Fig. 40 zeigt eine Darstellung eines Teils des Programmbausteins MM011;

Fig. 41 zeigt eine Darstellung eines Teils der Liste des Quellcodes für ein automatisch erstelltes Anwendungsprogramm.

Fig. 1 zeigt als eine Ausführungsform dieser Erfindung einen automatischen Programmgenerator, dessen Generator im wesentlichen besteht aus einer Zentraleinheit 1 und Arbeitsplätzen WS1–WSn. Während der automatische Programmgenerator aus Fig. 1 auf einem IBM AS/400 System betrieben werden kann, kann ein ähnlicher Generator auch auf Computern jeder anderen Marke betrieben werden.

Die Zentraleinheit 1 besteht aus einem zentralen Prozessor 2, einem Plattspeicher 3, einem Hauptspeicher 4 und wahlweise einem Drucker 5, der an die Zentraleinheit 1 angeschlossen ist. Der Arbeitsplatz WS ist mit einer Abfragestation zur Datenein- und -ausgabe und u. a. mit einem CRT-Bildschirm 6 und einer Tastatur 7 ausgerüstet, wenn notwendig, kann ein Drucker 8 an der Ausgabestation angeschlossen werden.

Bei dem in Fig. 1 dargestellten automatischen Programmgenerator werden die Quell- und Objektcodes eines Anwendungsprogramms dadurch automatisch erstellt, daß der Anwender am Arbeitsplatz WS im Dialog den Erstellungsvorgang der physikalischen Datei, der Bildschirmdateien, der Formblattdateien und der Programmdateien durcharbeitet. Des weiteren wird eine Liste des Quellcodes, die genau den erzeugten Objektcodes entspricht, sowie die verschiedenen Entwurfsdokumente, die genau den Objektcodes entsprechen (Datensatz-Entwurfsdokument, Bildschirmdatei-Entwurfsdokument, Entwurfsdokument) ebenfalls als Ausdruck ausgegeben. Die verwendete Programmiersprache ist keinen besonderen Einschränkungen unterworfen. Es können daher verschiedene Sprachen, wie die RPG-Sprache, die C-Sprache, die Cobol-Sprache usw. einge-

setzt werden. In der folgenden Beschreibung der Ausführungsform ist die Erzeugung eines Anwendungsprogramms in der RPG (report programm generator)-Sprache erklärt.

Bei dem in Fig. 1 dargestellten automatischen Programmgenerator werden zur Implementierung der obengenannten Verarbeitungen alle Computerverarbeitungsschritte in 36 verschiedenen Lösungen eingeteilt und die Programme zur Implementierung dieser Lösungen werden in 34 unterschiedlichen Programmsteine im Plattspeicher 3 gespeichert. Jeder der 34 verschiedenen Programmsteine stellt ein Quellprogramm in teilweise fertiggestelltem Zustand dar und besteht aus einem individuellen Anteil B, der für jede Anwendung verändert wird und einem Grundkörper A, der nicht verändert wird (Fig. 2). Der individuelle Anteil B wird bestimmt durch die Information, die über die Tastatur 7 während des Entwurfstadiums zur Erstellung des physikalischen Dateientwurfs, des Bilddateientwurfs, des Buchungsdateientwurfs oder des Programm-entwurfs eingegeben wurde.

Da jeder Programmstein ein hochwertiges Programm darstellt, welches von einem ausgebildeten Programmierer erstellt wurde und der Entwurf des Anwendungsprogramms im Dialog am Arbeitsplatz WS implementiert wird, ermöglicht der automatische Programmgenerator dieser Erfindung sogar einem Novizen auf einfache und schnelle Weise ein Anwendungsprogramm hoher Qualität zu erstellen.

Des weiteren kann, da alle Programme durch unterschiedliche Kombinationen von 34 Programmsteinen aufgebaut sind, das Format der unterschiedlichen Entwurfsdokumente standardisiert werden, zusätzlich können unterschiedliche Entwurfsdokumente in teilweise fertiggestelltem Zustand im Plattspeicher 3 gespeichert werden. Aus diesem Grund können, sofern ein Anwendungsprogramm einmal fertig gestellt wurde, die unterschiedlichen Entwurfsdokumente, welche genau mit dem Programm übereinstimmen, ausgedruckt werden. Da diese Unterlagen im Format standardisiert wurden, kann jeder einfach und genau den Inhalt der Verarbeitung des Anwendungsprogramms erfassen. In der Vergangenheit wurden diese unterschiedlichen Entwurfsdokumente nicht dem Anwender geliefert, oder entsprachen, falls sie geliefert wurden, nicht genau dem Programm. Da bei dem erfindungsgemäßen Programmgenerator Entwurfsdokumente, die genau mit dem Anwendungsprogramm übereinstimmen, zur Verfügung stehen, kann jedermann auf einfache Weise das Anwendungsprogramm überarbeiten oder modifizieren, so daß der wirtschaftliche Wert des an den Anwender ausgelieferten Anwendungsprogramms weiter erhöht wird.

Das gewünschte Anwendungsprogramm kann also aus jedem der 36 unterschiedlichen Bausteine oder aus einer Kombination dieser aufgebaut werden. Die 36 unterschiedlichen Bausteine werden grob eingeteilt in folgende Kategorien:

1. Systemeingabegenerierung (Entwurfsmuster 1–7),
2. Systemabfragegenerierung (Entwurfsmuster 8–14),
3. Hauptwartungsgenerierung (Entwurfsmuster 15–22),
4. Abfragenstergenerierung (Entwurfsmuster 23–26),
5. Druckbeleggenerierung (Entwurfsmuster 27–34) und

6. Änderungsdienstgenerierung (Entwurfsmuster 35–36).

1. Systemeingabegenerierung (Entwurfsmuster 1–7)

Mit dem Ausdruck "Systemeingabegenerierung" wird eine Generierung bezeichnet, die eine Übergabedatei betrifft, in der Daten zur Aktualisierung der Stammdatei gesammelt werden. Entsprechend der Art der bearbeiteten Daten wird diese Systemeingabegenerierung in sieben Generierungsarten (Entwurfsmuster 1–7) unterteilt und diese Generierungen der Entwurfsmuster 1–7 werden durch die Programmsteine ET010, ET011, ET050, ET051, ET052, ET150 bzw. ET153 implementiert (Fig. 3).

Wie in Fig. 3 gezeigt, ist die Systemeingabegenerierung in die Entwurfsmuster 1–5 und die Entwurfsmuster 6–7 aufgeteilt. Die Entwurfsmuster 1–5 enthalten Bausteine, die zur Verarbeitung von Daten aus Belegen geeignet sind, während die Entwurfsmuster 6–7 Bausteine sind, die zur Verarbeitung von editierbaren Daten geeignet sind.

Mit dem Ausdruck "Daten aus Belegen" wird eine Gruppe von Daten bezeichnet, die durch ihre Belegnummer bestimmt sind, und zwar in der Weise, daß die Daten im Spezifikationsteil den Zeilennummern zugeordnet werden und in einer Übergabedatei eingetragen werden. Ein beispielhafter Fall sind die Daten aus einem Einkaufsbeleg, der in Fig. 4 gezeigt ist. Dieser Einkaufsbeleg 11-1 bis 11-3 besteht aus einem Spezifikationsteil 11b, der den "Einkaufsgegenstand", sowie dem "Einzelpreis" und die "Einkaufsmenge" angibt sowie einer Kopfzeile 11a, die Informationen enthält, welche den Daten im Spezifikationsanteil 11 gemeinsam sind. Im Fall von Fig. 4 enthält die Information in der Kopfzeile 11a den "Lieferanten", das "Kaufdatum" und den "Einkäufer".

Die Belegnummer und die Zeilennummer werden nun genauer erklärt. Die drei Einkaufsbelege, die in Fig. 4 gezeigt sind, werden durch ihre Belegnummern 0001-0003 erkannt und die Daten aus dem Spezifikationsteil 11b jedes Belegs werden Zeilennummern zugeordnet und in einer Übergabedatei gespeichert. Diese Daten können daher für denselben "Einkaufsgegenstand" verarbeitet werden, auch wenn sie sich im "Einzelpreis" unterscheiden. Die Entwurfsmuster 1 bis 5 werden jetzt an Hand des Einkaufsbelegs 11 aus Fig. 4 beschrieben.

[Entwurfsmuster 1,2] Programmsteine ET010, ET011

Das Programm der Entwurfsmuster 1, 2 wird durch Einfügen des individuellen Teils B (Fig. 2) in die Programmsteine ET010 und ET011 durch Informationseingabe am Arbeitsplatz WS vervollständigt. Um die Daten der Belege in einer Übertragungsdatei einzutragen werden vorzugsweise die Entwurfsmuster 1–5 verwendet, ist die Informationsmenge in der Kopfzeile 11a groß, sind die Entwurfsmuster 1, 2 besonders geeignet. Wie in Fig. 5A gezeigt, werden die Daten der Kopfzeile 11a in einer Kopfzeilendatei 12 gespeichert und die Daten im Spezifikationsteil 11b werden in einer Spezifikationsdatei 13 gespeichert (Fig. 5B).

Nimmt man die Kopfzeilendatei aus Fig. 5A als Beispiel, hat der Einkaufsbeleg 11 aus Fig. 4 die Felder "Belegnummer" 12a, "Datum" 12b, "Lieferant" 12c, "Einkäufer" 12d, "Gesamtbetrag" 12e. Z.B. werden dann als

Suchbegriffe für eine Datensuche die Posten "Belegnummer", "Datum" und "Lieferant" eingesetzt.

Auf der anderen Seite hat z. B. die Spezifikationsdatei 13 die Felder "Belegnummer" 13a, "Datum" 13b, "Lieferant" 13c, "Zeilennummer" des Belegs 13d, "Einkaufsgegenstand" 13e, "Einzelpreis" 13f und "Einkaufsmenge" 13g. Hier können z. B. "Belegnummer", "Datum", "Lieferant" und "Zeilennummer" des Belegs als Suchbegriffe für eine Datensuche eingesetzt werden. Da die "Zeilennummer" des Belegs gespeichert und damit ein Suchbegriff zur Datensuche ist, ist es möglich, jeden einzelnen Datensatz entsprechend durch die Zeilennummer anzugeben und eine Löschung oder ein Kopieren zeilenweise durchzuführen.

Unter Bezugnahme auf das Flußdiagramm aus Fig. 6 wird die Verarbeitung des Entwurfsmusters 1 (Programmbaustein ET010) nun kurz beschrieben. Während das Entwurfsmuster 1 die Verarbeitung von Registrierung, Überarbeitung oder Löschung von Daten aus Belegen implementiert, werden jetzt die Vorgänge eines "Programms zur Verarbeitung von Einkaufsbelegen", das die Daten des Einkaufsbelegs 11 aus Fig. 4 verarbeitet, als eine Ausführungsform des Entwurfsmusters 1 erklärt.

Wird das "Programm zur Verarbeitung von Einkaufsbelegen" im Menü (FT1) ausgewählt, wird eine Maske für die Eingabe von Suchbegriffen für die Kopfzeilendatei 12 am Arbeitsplatz WS (ST2) angezeigt. Da der anfängliche Funktionsmodus des "Programms zur Verarbeitung von Einkaufsbelegen" der Registrationsmodus ist, wird zunächst die Neuregistrierung der Daten aus dem Einkaufsbeleg erklärt.

Im Schritt ST2 wird auf dem Bildschirm die Maske zur Informationseingabe in die Spalten "Belegnummer" 12a, "Datum" 12b und "Lieferant" 12c angezeigt. Der Anwender gibt die entsprechenden Informationen (im weiteren ID-Codes genannt) unter Verwendung von Ziffern und Symbolen ein. Je nach Aufbau des individuellen Anteils des Programmgerüsts ET010 wird "Belegnummer" automatisch zugeordnet, in diesem Fall ist die Eingabe der "Belegnummer" nicht notwendig.

Nachdem die Posten "Datum", "Lieferant" usw. am Arbeitsplatz eingegeben wurden, wird eine Maske zur Eingabe weiterer Informationen angezeigt, die nicht die Suchbegriffe der Kopfzeile 12 betreffen. In diesem Beispiel muß der "Einkäufer" eingegeben werden, der Anwender gibt darauf den entsprechenden ID-Code für den "Einkäufer" ein. Wird während des Schritts ST3 die Funktionstaste F12 gedrückt, kehrt die Programmfolge zum Schritt ST2 zurück, so daß die Maske zur Eingabe der Suchbegriffe in der Kopfzeile 12 angezeigt wird.

Wird auf der anderen Seite während des Schritts ST3 die "Enter"-Taste am Arbeitsplatz gedrückt, geht die Programmfolge zum Schritt (ST4) weiter zur Eingabe der Daten aus dem Spezifikationsteil 11b des Einkaufsbelegs 11, so daß der Anwender die entsprechenden ID-Codes für "Einkaufsgegenstand", "Einzelpreis" und "Einkaufsmenge", eingeben kann. Nach vollendeter Eingabe der Daten des Spezifikationsteils 11b des Einkaufsbelegs 11 werden am Arbeitsplatz die Multiplikation von "Einzelpreis" und "Einkaufsmenge" sowie die Addition der Produkte zusammen mit den Ergebnissen der entsprechenden Rechenoperationen angezeigt (ST5). Zur selben Zeit wird der Anwender aufgefordert, die eingegebenen Daten zu bestätigen (ST5).

Drückt der Anwender die "Enter"-Taste werden die in der Maske angezeigten Daten in der Kopfzeilendatei 12 und den entsprechenden Bereichen der Spezifikations-

datei 13 gespeichert (ST6). Falls notwendig, kann der Inhalt der Stammdaten (z. B. einer Stammdaten zur Inventur) entsprechend aktualisiert werden (ST7). Wird die Funktionstaste F12 während der Eingabe der Spezifikationsdaten (ST4) oder während der Bestätigungsaufrichtung gedrückt, kehrt die Programmfolge zur Verarbeitung des Schritts ST3 zurück. Wird die Funktionstaste F3 während der Verarbeitung der Schritte ST2–ST5 gedrückt, wird wieder das Menü des Schritts ST1 angezeigt.

Im Fall eines Programms mit dem Entwurfsmuster 1 kann, wie bei den anderen Entwurfsmustern auch, eine Verarbeitung unter Verwendung von Fenstern durchgeführt werden. Unter dem Ausdruck "Verarbeitung mit Fenstern" wird eine Verarbeitung verstanden, die die Verarbeitung der Schritte ST2–ST5 durch Gebrauch eines Teils des CRT-Bildschirms (Fensterfläche W) unterstützt (Fig. 7). Die Verarbeitung mit Fenstern besteht aus einem grundlegenden Teil und einem optionalen Teil, wobei der optionale Teil selektiv zugefügt oder unterdrückt wird indem die individuellen Teile des Programmbausteins ET010, ET011 entsprechend eingerichtet werden. Verarbeitungen mit Fenstern, die optional ausgewählt werden können sind in den Programmbausteinen WN010, WN011, WN012 und WN020 vorbereitet, welche als nächstes beschrieben werden.

In den Entwurfsmustern 1, 2 sind der Moduswechsel über die Funktionstaste F5 und die zeilenweise Verarbeitung über die Funktionstaste F9 grundlegende Verarbeitungsschritte, während die Code-Suche über die Funktionstaste F4 eine optionale Verarbeitung ist, die durch die Einrichtung des individuellen Teils des Programmbausteins ausgewählt wird.

Mit dem Ausdruck "Moduswechsel" ist eine Verarbeitung gemeint, mit der vom aktuellen Verarbeitungsmodus des Programms in den Modus der Datenregistrierung, den Überarbeitungsmodus oder den Löschungsmodus gewechselt werden kann, wird die Funktionstaste F5 während der Verarbeitung des Schritts ST2 gedrückt, wird ein Moduswechsel durch Verarbeitung im Fenster W durchgeführt. Nimmt man z. B. an, daß in dem in Fig. 7 gezeigten Status eine "2" eingegeben wird, wechselt der aktuelle Modus des "Programms zur Verarbeitung von Einkaufsbelegen" in den Überarbeitungsmodus, und während der Verarbeitung der Schritte ST2 bis ST5 werden die in den Übergabedateien 12 und 13 gespeicherten Daten überarbeitet (ST6). Im Überarbeitungsmodus werden die Daten des Einkaufsbelegs, die in den Übergabedateien 12, 13 gespeichert wurden, während der Verarbeitung der Schritte ST2 und ST4 am Arbeitsplatz angezeigt.

Wird während des in Fig. 7 dargestellten Status "3" eingegeben, wechselt der aktuelle Modus des "Verarbeitungsprogramms für Einkaufsbelege" in den Löschungsmodus und die im Verarbeitungsschritt ST2 erhaltenen Daten des Einkaufsbelegs werden aus den Übergabedateien 12, 13 gelöscht (ST6). Im Löschungsmodus werden nur die gespeicherten Daten des Einkaufsbelegs während der Verarbeitung des Schritts ST4 angezeigt, die Verarbeitung des Schritts ST5 wird nicht ausgeführt, sondern übersprungen.

Der mit der Funktionstaste F9 verbundene Ausdruck "zeilenweise Verarbeitung" bedeutet eine Verarbeitung, die unter der Verwaltung jedes einzelnen Datensatzes in der Spezifikationsdatei 13 nach der "Zeilennummer" durchgeführt wird, typischerweise ist mit dieser Verarbeitung ein Vorgang verbunden, bei dem die Daten aus einer bestimmten Zeile an eine andere Stelle mit einer

anderen Zeilennummer kopiert werden.

Der mit der Funktionstaste F4 verbundene Begriff "Code-Suche" bedeutet eine Verarbeitung, um einen einzugebenden ID-Code aufzufinden. Wie oben erwähnt, muß die gesamte notwendige Information in den Schritten ST2 bis ST5 unter Benutzung von ID-Codes eingegeben werden. Sind daher die ID-Codes für "ABC Industrial Co.", "Druckerboard" und "Herr Sato" nicht bekannt, können diese durch Drücken der Funktionstaste F4 ermittelt werden.

Während der Inhalt des "Programms zur Verarbeitung von Einkaufsbelegen", das aus dem Programmabaustein ET010 (Entwurfsmuster 1) aufgebaut ist, oben beschrieben wurde, stimmt der Programmabaustein ET011 (Entwurfsmuster 2) mit diesem im wesentlichen überein, der einzige Unterschied besteht darin, daß kein Additionsschritt im Schritt ST5 ausgeführt wird. Der Programmabaustein ET011 (Entwurfsmuster 2) eignet sich daher zur Verarbeitung der Registrierung, Überarbeitung oder Löschung von Daten aus Belegen sowie in den Fällen, bei denen die Gesamtsumme o. dgl. am Arbeitsplatz nicht angezeigt werden muß.

[Entwurfsmuster 3—5] Programmabusteine ET050,
ET051, ET052

Die Entwurfsmuster 3—5 sind dadurch gekennzeichnet, daß keine Kopfzeile vorhanden ist. Sie eignen sich, Daten aus der Kopfzeile 11a oder auch aus dem Spezifikationsteil 11b in einer Spezifikationsdatei 14 abzuspeichern, wie dies z. B. mit den Daten aus Fig. 4 gezeigt ist. Ein solcher Fall ist in Fig. 8 dargestellt. Die Spezifikationsdatei 14 besteht aus einem Kopfzeilenteil 14A und einem Spezifikationsteil 14B, wobei der Kopfzeilenteil 14A die Felder "Belegnummer" 14A1, "Datum" 14A2, "Lieferant" 14A3 und "Einkäufer" 14A4 aufweist und der Spezifikationsteil 14B die Felder "Zeilennummer" 14B1, "Einkaufsgegenstand" 14B2, "Einzelpreis" 14B3, "Einkaufsmenge" 14B4 und "Gesamtbetrag" 14B5.

Die Verarbeitung des Entwurfsmusters 3 (Programmabaustein ET050) erfolgt wie im Flußdiagramm in Fig. 9 gezeigt. Ein Vergleich von Fig. 9 mit Fig. 6 zeigt deutlich, daß der Ablauf der Schritte ST1' bis ST6' in Fig. 9 im wesentlichen übereinstimmt mit der Verarbeitung der Schritte ST1 bis ST6 in Fig. 6 und sich das Entwurfsmuster 3 vom Entwurfsmuster 1 nur darin unterscheidet, daß keine Kopfzeile vorhanden ist.

Das Entwurfsmuster 4 (Programmabaustein ET051) führt nahezu dieselben Verarbeitungen aus, wie das in Fig. 9 gezeigte Entwurfsmuster 3, der Programmabaustein ET051 unterscheidet sich jedoch vom Programmabaustein ET050 darin, daß keine Addition in der Verarbeitung im Schritt ST5' in Fig. 9 durchgeführt wird.

Entwurfsmuster 5 (Programmabaustein ET052) führt ebenfalls nahezu dieselben Verarbeitungen aus wie das in Fig. 9 gezeigte Entwurfsmuster 3, der Programmabaustein ET052 unterscheidet sich vom Programmabaustein ET050 dadurch, daß die Verarbeitung des Schritts ST3' in Fig. 9 fehlt. Das Entwurfsmuster 5 wird daher gewählt, wenn alle Daten im Teil 14A der Kopfzeile der Spezifikationsdatei 14 Suchbegriffe sind. Ein solcher Fall liegt z. B. vor, wenn bezogen auf den Spezifikationsteil 14A in Fig. 8, die "Belegnummer" 14A1, das "Belegdatum" 14A2 und der "Lieferant" 14A3 als Suchbegriffe für eine Datensuche genutzt werden und das Feld "Einkäufer" nicht zur Verfügung steht.

[Entwurfsmuster 6—7] Programmabusteine ET150,
ET153

Die Entwurfsmuster 6—7 sind zum Editieren (Registration und Überarbeitung) oder zum Löschen von editierbaren Daten geeignet. Der Ausdruck "editierbare Daten" bezieht sich auf Daten, die nicht einer Belegnummer oder Zeilennummer zugeordnet sind, oder mit anderen Worten auf solche Daten, die als Spezifikationsdaten als Suchbegriff zur Datensuche eingesetzt werden können. Ein typisches Beispiel für editierbare Daten sind Daten, die anfallen wenn ein Kundendienstmitarbeiter das Tagesprotokoll von Wartung und Kontrolle in eine Übergabedatei einträgt. Wie in Fig. 10 dargestellt, muß während "Herr Nakao", ein Kundendienstmitarbeiter, Dienstleistungen wie "Inspektion" oder "Wartung" für "RCC", "NTT" oder andere Kunden erbringt, ein Protokoll seiner täglichen Tätigkeiten in einer Übergabedatei aufgenommen werden, die Kosten für jede Monatsperiode für jeden Kunden aufaddiert werden und eine Rechnung erstellt werden. In derartigen Fällen wird für die Daten des Kundendienstprotokolls, die in der Übergabedatei gespeichert werden, keine Belegnummer oder Vergleichbares benötigt, noch müssen die Spezifikationsdaten mit Zeilennummern versehen werden. Die Anwendung der Entwurfsmuster 6 und 7 ist daher angebracht.

In Fig. 11 ist die Übergabedatei 15 gezeigt, die zur Speicherung des Kundendienstprotokolls eines Kundendienstmitarbeiters dient. Diese Spezifikationsdatei besteht aus einem Kopfzeilenteil 15A und einem Spezifikationsteil 15B, wobei der Kopfzeilenteil 15A die Felder "Datum" 15A1 und "Mitarbeiter" 15A2 enthält und der Spezifikationsteil 15B die Felder "Kunde" 15B1, "Dienstleistung" 15B2, "Zeit in Stunden" 15B3, "Preis/Einheit" 15B4 und "Gesamtbetrag" 15B5. "Datum" 15A1, "Mitarbeiter" 15A2 und "Kunde" 15B1 werden hier als Suchbegriffe für eine Datensuche angenommen.

Die Wirkungsweise eines "Programms zur Registrierung eines Kundendienstprotokolls", das aus dem Entwurfsmuster 6 (Programmabaustein ET150) aufgebaut ist, wird im weiteren unter Bezugnahme auf das Flußdiagramm in Fig. 12 beschrieben.

Wird das "Programm zur Registrierung eines Kundendienstprotokolls" im Menü angewählt (ST10), wird eine Maske zur Eingabe der Suchbegriffe des Kopfzeilenteils 15A an der Arbeitsstelle angezeigt (ST11). Da "Datum" und "Mitarbeiter" im obigen Beispiel Suchbegriffe darstellen, gibt der Anwender die entsprechenden Zahlen und ID-Codes ein. Es werde angenommen, daß das Kundendienstprotokoll vom 24. September, das dem Mitarbeiter "Nakao" zugeordnet ist, eingegeben wird (ST11).

Als nächstes wird eine Maske zur Eingabe anderer Information als die der Suchbegriffe des Kopfzeilenteils angezeigt (ST12). Bei diesem Beispiel muß im Schritt ST12 keine Information eingegeben werden, es kann jedoch der Name der Abteilung der "Nakao" zugehört, eingegeben werden. Wird während des Schritts ST12 die Funktionstaste F12 gedrückt, kehrt die Programmfolge zum Schritt ST11 zur Eingabe von "Datum" und "Mitarbeiter" zurück.

Wird auf der anderen Seite die "Enter"-Taste während des Schritts ST12 gedrückt, wird eine Maske zur Eingabe der Daten des Spezifikationsteils 15B der Spezifikationsdatei 15 angezeigt. In diesem Beispiel wird, da das Kundendienstprotokoll vom 21.—23. September für den Kundendienstmitarbeiter "Nakao" bereits eingetragen

wurde, die registrierte Information am Arbeitsplatz angezeigt. Der Anwender gibt daher nach dieser angezeigten Information die ID-Codes und die Zahlen ein, welche dem "Kunden" der "Dienstleistung", der "Zeit in Stunden" und dem "Preis/Einheit" entsprechen (ST13). Falls es notwendig sein sollte, die registrierten Daten zu kontrollieren oder zu löschen, wird die Korrektur der Daten an diesem Punkt durchgeführt.

Nach Abschluß der Dateneingabe in den Spezifikationsteil 15B wird die Rechenoperation zur Multiplikation von "Preis" und "Zeit in Stunden" sowie die Addition der Leistungen durchgeführt und die Ergebnisse der entsprechenden Rechenoperationen am Arbeitsplatz angezeigt (ST14). Zur selben Zeit wird eine Sortierung nach dem ID-Code des "Kunden" durchgeführt und die Informationen, welche den Kundendienst des Kundendienstmitarbeiter "Nakao" zwischen dem 21. und 24. September betrifft nach den Kunden sortiert und am Arbeitsplatz angezeigt. Außerdem wird der Anwender aufgefordert, die eingegebenen Daten zu bestätigen (ST14).

Drückt der Anwender die "Enter"-Taste werden die Daten in der Spezifikationsdatei 15 gespeichert (ST15). Falls eine Stammdatei existiert, die aktualisiert werden muß, wird der Inhalt der Stammdatei entsprechend aktualisiert (ST16).

Wird während der Eingabe der Spezifikationsdateien (ST13) und der Anzeige der Bestätigungsaufruf (ST14) die Funktionstaste F14 gedrückt, kehrt die Programmabfolge zur Verarbeitung des Schritts ST12 zurück. Wird die Funktionstaste F3 während der Verarbeitung der Schritte ST11 bis ST14 gedrückt, kehrt die Programmabfolge zur Menümaske (ST10) zurück.

Die Funktionsweise des "Programms zur Erstellung eines Kundendienstprotokolls" im Editiermodus wurde bereits beschrieben. Ist der aktuelle Modus der Löschungsmodus, werden die Daten des ausgewählten "Datum" für den entsprechenden Kundendienstmitarbeiter als Ganzes gelöscht. Die Arbeitsweise mit Fenstern wird identisch wie beim Entwurfsmusters 1 durchgeführt. Ein Moduswechsel mit Hilfe der Funktionstaste F5 und eine zeilenweise Verarbeitung mit der Funktionstaste F9 sind daher grundlegende Verarbeitungen, während die Suche nach Codes mit der Funktionstaste F4 eine optionale Verarbeitung ist, welche durch Einstellung des individuellen Teils des Programmbausteins ausgewählt wird.

Das Entwurfsmuster 7 (Programmbaustein ET153) stimmt im wesentlichen mit dem Entwurfsmuster 6 überein, der einzige Unterschied besteht darin, daß der Verarbeitungsschritt ST12 fehlt und die Additionsoperation im Schritt ST14 nicht durchgeführt wird. Sind also alle Daten im Kopfzeileteil 15A Suchbegriffe und wird eine Additionsoperation für den Spezifikationsteil 15B nicht benötigt, ist das Entwurfsmuster 7 die geeignete Wahl.

2. Systemabfrageverarbeitung (Entwurfsmuster 8–14)

Die Entwurfsmuster 8–14 sind in den Fällen geeignet, in denen ein Befehl zum Informationsaufruf vom Arbeitsplatz an den Plottenspeicher ausgeführt wird.

3. Diese Muster sind in zwei Hauptgruppen eingeteilt, nämlich die Entwurfsmuster 8–11, welche geeignet sind, sollte die gesuchte Information nicht genau angegeben werden können, und die Entwurfsmuster 12–14, welche geeignet sind, wenn die gesuchten Daten ziemlich genau angegeben werden können.

Die Entwurfsmuster 8–11 werden durch Kombination des Programmbausteins SK010 mit den Programmbausteinen SK020, SK021, SK040 und SK041 erstellt, während die Entwurfsmuster 12–14 aus den Programmbausteinen SK050, SK051 und SK060 erstellt werden. Es gilt also:

Entwurfsmuster 8 = SK010 + SK020
Entwurfsmuster 9 = SK010 + SK021
Entwurfsmuster 10 = SK010 + SK040
Entwurfsmuster 11 = SK010 + SK041
Entwurfsmuster 12 = SK050
Entwurfsmuster 13 = SK051
Entwurfsmuster 14 = SK060.

Die Programmbausteine SK020, SK021, SK050 und SK051 sind zur Datensuche in einer Übergabedatei geeignet, während sich die Programmbausteine SK040, SK041 und SK060 zur Datensuche in einer Stammdatei eignen.

[Entwurfsmuster 8, Entwurfsmuster 9]

Programmbausteine SK010 + SK020, SK021)

Wie in Fig. 3 angegeben, besteht der Programmbaustein SK010 aus einer Verarbeitung (ST22) zur Eingabe der Suchbedingungen für die gesuchten Daten und einer Verarbeitung (ST23–ST25) zur Angabe des Suchprogramms. Während die Entwurfsmuster 8, 9 Kombinationen des Programmbausteins SK010 mit den Programmbausteinen SK020 und SK021 darstellen, eignen sich die Programmbausteine SK020 und SK021 zur Anzeige von Daten aus Übergabedateien, welche durch die Programme der Entwurfsmuster 1–7 aufgefunden wurden.

Unter der Annahme, daß ein "Suchprogramm für Einkaufsbelege" aus einer Kombination der Programmbausteine SK010 und SK020 aufgebaut ist, wird die Wirkung der Entwurfsmuster 9 und Entwurfsmuster 10 nun kurz beschrieben (Fig. 13). Es wird ferner angenommen, daß die Daten des Einkaufsbelegs 11 (Fig. 4) bereits wie in Fig. 5 gezeigt, in der Kopfzeiledatei 12 und der Spezifikationsdatei 13 gespeichert sind.

Wird das "Suchprogramm für Einkaufsbelege" im Menü ausgewählt (ST21), wird zunächst am Arbeitsplatz eine Maske zur Eingabe der Suchbedingungen angezeigt (ST22), zu diesem Zeitpunkt muß der Anwender die Bedingungen für die gesuchten Daten eingeben. Es wird angenommen, daß der Anwender einen Datumsbereich von z. B. "3. Oktober bis 10. Oktober 1994" eingibt (ST22).

In der Folge wird in der Kopfzeiledatei 12 gesucht, wobei obengenannter Datumsbereich als Suchbegriff dient und als Ergebnis alle entsprechenden Daten auf dem CRT-Bildschirm des Arbeitsplatzes angezeigt werden (ST23; Fig. 14). Wie in Fig. 14 gezeigt, enthält die am Arbeitsplatz angezeigte Liste eine Spalte "Funktionscode". Mit dem Ausdruck "Funktionscode" wird ein Code bezeichnet, welcher bestimmt, welches Suchprogramm während der Ausführung der Suchverarbeitung zur Ausgabe der gesuchten Daten benutzt werden soll. Hier sind die Funktionscodes "1" und "2" den Suchprogrammen "PRG1" und "PRG2" zugeordnet. In diesem Beispiel wird angenommen, daß das Suchprogramm "PRG1" aus dem Programmbaustein SK020 besteht, während das Suchprogramm "PRG2" aus dem Programmbaustein SK021 aufgebaut ist. Nach Festlegung des Funktionscodes wird die Verarbeitung entweder des Suchprogramms 1 oder des Suchprogramms 2 gestartet (ST25).

Es wird angenommen, daß mit Hilfe der Cursorbewegung die zweite Zeile der in Fig. 14 dargestellten Maske ausgewählt wird und durch den Funktionscode "1" das Suchprogramm "PRG1" ausgewählt wird. Als nächstes wird die Spezifikationsdatei 13 durchsucht, wobei die Belegnummer "1", das Datum "3.10.1994" und "ABC Industrials Co." als Suchbegriffe eingesetzt werden, und die entsprechenden Dateien am Arbeitsplatz angezeigt. Das oben Geschilderte ist der Inhalt der Verarbeitung des Entwurfsmusters 8.

Der im wesentlichen gleiche Ablauf trifft zu, wenn mit dem Funktionscode das Suchprogramm "PRG2" ausgewählt wird. Der einzige Unterschied besteht darin, daß eine Multiplikation von Einzelpreis und Einkaufsmenge für jeden Einkaufsposten, sowie eine Addition der Produkte durchgeführt wird und die Ergebnisse am Arbeitsplatz angezeigt werden. Das oben Geschilderte ist der Inhalt der Verarbeitungen des Entwurfsmusters 9. Dieser Programmbaustein SK021 (Entwurfsmuster 9) unterscheidet sich vom Programmbaustein SK020 (Entwurfsmuster 8) nur darin, daß die erwähnten Operationen durchgeführt und die Ergebnisse angezeigt werden.

[Entwurfsmuster 10] Programmbausteine SK010 + SK040

Das Entwurfsmuster 10 besteht aus einer Kombination des Programmbausteins SK010 und des Programmbausteins SK040. Programmbaustein SK040 eignet sich zur Anzeige der in den Entwurfsmustern 15–22 gespeicherten Daten der Stammdatei. Z.B. wird im in Fig. 18 dargestellten Fall eine Kundenstammdatei gesucht, als Suchbedingung wird angenommen, daß der Kundencode "0020-0030" lautet (ST22 in Fig. 13), darauf wird, wie in Fig. 15 dargestellt, eine Liste der entsprechenden Firmennamen angezeigt (ST23). Der Cursor wird auf die Position des gesuchten Kunden bewegt und der Funktionscode eingegeben (ST24), daraufhin wird die Verarbeitung des durch den Funktionscode bestimmten Suchprogramms gestartet und die entsprechende Information am Arbeitsplatz angezeigt (ST25). Beim Entwurfsmuster 10 ist dieses Suchprogramm aus dem Programmbaustein SK040 aufgebaut, während das Entwurfsmuster die Verarbeitung zur Anzeige der Daten der Stammdatei am Arbeitsplatz ausführt.

[Entwurfsmuster 11] Programmbaustein SK010 + SK041

Entwurfsmuster 11 besteht aus einer Kombination der Programmbausteine SK010 und SK041 und führt im wesentlichen dieselben Verarbeitungen zur Anzeige der Stammdatei am Arbeitsplatz aus wie das Entwurfsmuster 10. In Fig. 16 ist in einem Flußdiagramm ein Überblick der Verarbeitungen des Programmbausteins SK041 dargestellt. Der Unterschied zwischen diesem Programmbaustein und dem Programmbaustein SK040 besteht darin, daß die Daten auf zwei Bildschirmseiten am Arbeitsplatz angezeigt werden. Das Entwurfsmuster 11 ist daher dann geeignet, wenn ein Eintrag sehr viele Daten enthält und nicht auf einer einzelnen Bildschirmseite dargestellt werden kann. Wie in Fig. 16 gezeigt, werden die Funktionstasten F8 und F7 zum Hin- und Herspringen zwischen einer ersten Bildschirmseite und einer zweiten Bildschirmseite benutzt.

[Entwurfsmuster 12, 13, 14] Programmbaustein SK050, SK051, SK060

Die Entwurfsmuster 12–14 sind dadurch gekennzeichnet, daß eine Verarbeitung zur Anzeige von Listen, wie sie z. B. vom Programmbaustein SK010 ausgeführt wird, nicht durchgeführt wird. Während die Entwurfsmuster 8–11 für Situationen geeignet sind, in denen zu suchende Daten nicht genau angegeben werden können, sind die Entwurfsmuster 12–14 für Situationen geeignet, in denen die gesuchten Daten mehr oder weniger genau angegeben werden können. Die Entwurfsmuster 12 und 13 sind im wesentlichen gleichartig zu den Entwurfsmuster 8 und 9 mit dem Unterschied, daß die entsprechenden Daten nicht in Form einer Liste angezeigt werden und zur Bezugnahme auf den Inhalt einer Übergabedatei geeignet sind. Entwurfsmuster 14 ist nahezu identisch mit dem Entwurfsmuster 10 und sich zur Bezugnahme auf den Inhalt einer Stammdatei eignen.

10 Im Fall eines Programms mit dem Entwurfsmuster 12 (Programmbaustein SK050), wird, wie in Fig. 17 gezeigt, zunächst eine Maske zur Eingabe der Suchbegriffe am Arbeitsplatz angezeigt (ST42). Es wird z. B. angenommen, daß unter Bezugnahme auf die Spezifikationsdatei 15 aus Fig. 8 die "Belegnummer", das "Datum" und der ID-Code des "Lieferanten" eingegeben werden. Die Spezifikationsdatei 14 wird dann unter Verwendung dieser Suchbegriffe durchsucht und die Spezifikationsdaten der entsprechenden Belegnummer am Arbeitsplatz angezeigt. Wird während der Anzeige der gesuchten Spezifikationsdaten (ST43) die Funktionstaste F12 gedrückt, kehrt die Programmfolge zu Schritt ST42 zurück, wobei die eingegebenen Suchbegriffe gesichert werden. Unabhängig vom Zustand löscht ein Drücken der Funktionstaste F3 die Anzeige und das Menü wird wieder angezeigt (ST41). Eine Verarbeitung zur Codesuche mit der Funktionstaste F4 stellt eine optional auswählbare Fensterverarbeitung dar, und in jedem Anwendungsprogramm, für das diese Verarbeitung ausgewählt wurde, können während der Verarbeitung des Schritts ST42 ID-Codes gesucht werden.

Der Programmbaustein SK051 (Entwurfsmuster 13) führt im wesentlichen dieselben Verarbeitungen aus wie der Programmbaustein SK050, er unterscheidet sich von letzterem jedoch dadurch, daß eine Addition der gesuchten Daten ausgeführt und das Ergebnis angezeigt wird.

Programmbaustein SK060 (Entwurfsmuster 14), der im wesentlichen identisch mit dem Programmbaustein SK040 ist, eignet sich zur Anzeige von Daten aus der Stammdatei, die mit den Entwurfsmustern 15–22 gespeichert wurden. So wird z. B. nach Eingabe der Suchbegriffe die entsprechende Information aus der Stammdatei aufgerufen, und ein entsprechender Datensatz am Arbeitsplatz angezeigt.

3. Verarbeitungssystem zur Pflege von Stammdateien
(Entwurfsmuster 15–22)

Mit dem Ausdruck "Verarbeitungssystem zur Pflege von Stammdateien" wird eine Verarbeitung zur Registrierung, Überarbeitung, Löschung oder Anzeige eines einzelnen Datensatzes in der Stammdatei bezeichnet. Für das Programm zur Pflege von Stammdateien stehen die Entwurfsmuster 15–22 entsprechend der Größe eines Datensatzes zum selektiven Gebrauch zur Verfügung und die Programmbausteine MM010, NM011, MM020, MM021, MM030, MM031, MM050 und

MM051 entsprechen jeweils diesen Entwurfsmustern. So werden die Programmbausteine MM010 und MM011 zur Anzeige eines Datensatzes eingesetzt, für den eine Bildschirmseite ausreicht. Die Programmbausteine MM020 und MM021 werden eingesetzt zur Anzeige eines Datensatzes der zwei Bildschirmseiten benötigt, Programmbausteine MM030 und MM031 für einen Datensatz der drei Bildschirmseiten erforderlich und die Programmbausteine MM050 und MM051 für einen Datensatz der vier Bildschirmseiten erforderlich.

Für eine weitergehende Erklärung wird die "Kundenstammdaten" aus Fig. 18 als Beispiel für eine Stammdaten herangezogen. Diese Kundenstammdaten hat die Felder "Kundencode" 16A, "Firmenname" 16B, "Firmenart" 16C, "Adresse" 16D, "Telefonnummer" 16E und "Faxnummer" 16F, wobei der Kundencode als Suchbegriff für die Datensuche herangezogen wird.

[Entwurfsmuster 15] Programmbaustein MM010

An Hand des Flußdiagramms in Fig. 19 wird jetzt genauer auf den Inhalt der Funktionen des "Pflegeprogramm für Kundenstammdaten" eingegangen, welches aus dem Programmbaustein MM010 aufgebaut ist. Dieses Programm wurde zur Eintragung eines Datensatzes in die in Fig. 18 gezeigte Kundenstammdaten 17 entworfen.

Wird im Menü das Pflegeprogramm für Kundenstammdaten ausgewählt (ST51), erscheint am Arbeitsplatz eine Bildschirmmaske zur Eingabe der Suchbegriffe (ST52). Der Anwender gibt daher den ID-Code des Kunden ein (ST52). Es wird angenommen, daß der Funktionsmodus des Programms der "Registrierungsmodus" sei. Am Arbeitsplatz erscheint dann eine Bildschirmseite zur Eingabe von "Firmenname", "Firmenart", "Adresse", "Telefonnummer" und "Faxnummer". Der Anwender gibt darauf die benötigte Information ein (ST53) und drückt anschließend die "Enter"-Taste. Daraufhin wird eine Aufforderung zur Bestätigung angezeigt (ST54) und die gerade eingegebenen Daten werden, nachdem wiederum die "Enter"-Taste gedrückt wurde, zur Kundenstammdaten hinzugefügt (ST55).

Der "Registrierungsmodus" wurde bereits oben beschrieben. Wird die Funktionstaste F5 während des Schritts ST52 gedrückt, kann der Funktionsmodus des Programms mit Hilfe einer Fensterverarbeitung gewechselt werden. Wird die Taste (ST52) eingegeben, nachdem der Funktionsmodus des Programms in den "Überarbeitungsmodus" gewechselt wurde, wird der entsprechende einzelne Datensatz auf der Bildschirmseite angezeigt und in der Folge kann die entsprechende Position überarbeitet werden (ST53). Wird die "Enter"-Taste gedrückt, während die Aufforderung zur Bestätigung angezeigt wird (ST54), wird der Inhalt der Kundenstammdaten überarbeitet (ST55).

Im "Löschungsmodus" wird der angewählte einzelne Datensatz nach Drücken der Taste vollständig gelöscht. Im "Anzeigemodus" wird der angewählte einzelne Datensatz lediglich am Arbeitsplatz angezeigt, die Verarbeitungen der Schritte ST54 und ST55 werden übersprungen.

[Entwurfsmuster 16] Programmbaustein MM011

Das Entwurfsmuster 16 ist dadurch gekennzeichnet, daß während der Verarbeitung des Schritts ST52 in Fig. 19 eine Kopierfunktion eingesetzt werden kann, in den anderen Punkten ist es identisch mit dem Entwurfs-

muster 15. Der Ausdruck "Kopierfunktion" bezeichnet eine Funktion, mit der ein bereits abgespeicherter Datensatz in einen neuen Eintrag kopiert werden kann. Genauer gesagt, erscheint beim Entwurfsmuster 16 während der Verarbeitung des Schritts ST52 eine Bildschirmmaske zur Eingabe des "zu kopierenden Originaleintrags" (Fig. 20). Wird der Kundencode, z. B. "0001", in die Eingabespalte des "zu kopierenden Originaleintrags" eingegeben, wird ein einzelner Datensatz von "ABC Industrial Co." auf der Bildschirmseite am Arbeitsplatz angezeigt. Dieses Entwurfsmuster ist daher günstig, wenn die einzugebenden Daten den Daten von "ABC Industrial Co." ähneln, wenn also nur der Teil, welcher von den Daten in "ABC Industrial Co." abweicht, eingegeben werden muß.

[Entwurfsmuster 17, 18] Programmbaustein MM020, MM021

Das Entwurfsmuster 17 ist geeignet, wenn die Eingabe eines Datensatzes zwei Bildschirmseiten erfordert und ist dadurch gekennzeichnet, daß die Verarbeitung der Schritte ST53 und ST54 in Fig. 9 zweimal wiederholt wird. Das Entwurfsmuster 18 ist dadurch gekennzeichnet, daß eine Kopierfunktion verwendet werden kann und ist sonst identisch mit dem Entwurfsmuster 17.

[Entwurfsmuster 19, 20] Programmbaustein MM030, MM031

Das Entwurfsmuster 19 ist geeignet, wenn die Eingabe eines Datensatzes am Arbeitsplatz drei Bildschirmseiten erfordert und ist dadurch gekennzeichnet, daß die Verarbeitung der Schritte ST53 und ST54 in Fig. 19 insgesamt dreimal wiederholt wird. Entwurfsmuster 20 weist eine Kopierfunktion auf und ist sonst identisch mit Entwurfsmuster 19.

[Entwurfsmuster 21, 22] Programmbaustein MM050, MM051

Das Entwurfsmuster 21 ist geeignet, wenn die Eingabe eines Datensatzes am Arbeitsplatz fünf Bildschirmseiten erfordert und ist dadurch gekennzeichnet, daß die Verarbeitung der ST53 und ST54 in Fig. 19 insgesamt fünfmal wiederholt werden. Entwurfsmuster 22 weist eine Kopierfunktion auf und ist sonst identisch mit Entwurfsmuster 21.

4. Fensterverarbeitung des Abfragesystems (Entwurfsmuster 23—26)

Mit dem Ausdruck "Fensterverarbeitung des Abfragesystems" wird eine Suchverarbeitung bezeichnet, bei der das Fenster "W" (Fig. 7) eingesetzt wird, zur Ausführung der Entwurfsmuster 23—26 stehen die Programmbausteine WN010, WN011, WN012 und WN020 zur Verfügung. Die Verarbeitungen der Entwurfsmuster 22—26 sind unveränderliche Datensuchverarbeitungen, die während der Ausführung eines Aufrufprogramms mit der Funktionstaste F4 aufgerufen werden. Während sich die Programmbausteine WN010, WN011 und WN012 für Situationen eignen, in denen Daten in einer Stammdaten gesucht werden, ist der Programmbaustein WN020 geeignet, wenn Daten in einer Übergabedatei gesucht werden.

[Entwurfsmuster 23] Programmbaustein WN010

Das Programm des Entwurfsmusters 23 ist z. B. so aufgebaut, daß, falls während der Verarbeitung des Schritts ST52 des "Pflegeprogramms für Kundenstammdatei" (Fig. 19) die Funktionstaste F4 gedrückt wird, die "Kundenstammdatei" durchsucht wird, um den ID-Code des "Kunden" zu ermitteln. Im "Pflegeprogramm für Kundenstammdateien" muß der ID-Code des Kunden im Schritt ST52 eingegeben werden. Wird die Funktionstaste F4 gedrückt, wobei die Eingabespalte für die Eingabe des ID-Code leer ist, oder wird die Funktionstaste F4 gedrückt, wobei in die Spalte für die Eingabe des ID der Code eingetragen ist (ST52 und ST60), beginnt der Programmbaustein WN010 zu arbeiten (Fig. 21).

Beim Programmbaustein WN010 wird zunächst geprüft, ob ein Parameter vorhanden ist (ST61), und falls dieser nicht existiert, geht die Programmfolge weiter zu Schritt ST62 und falls er existiert, geht die Programmfolge zu Schritt ST63. Der Ausdruck "Parameter" wird im obigen Zusammenhang benutzt, um einen Startcode für die Suche zu bezeichnen, der durch Drücken der Funktionstaste F4 während der Ausführung eines Aufrufprogramms bestimmt wird; im vorliegenden Fall entspricht der während der Verarbeitung des Schritts ST52 des "Pflegeprogramms für Kundenstammdateien" eingegebene Kunden-ID-Code dem Startcode für die Suche.

Wird im "Pflegeprogramm für Kundenstammdateien" die Funktionstaste F4 gedrückt ohne daß ein Kunden-ID-Code eingegeben wurde, wird die Angabe im Schritt ST61 angefordert und es erscheint eine Fensteranzeige zur Eingabe der Startbedingungen für die Suche (ST62). Da im vorliegenden Beispiel jetzt der Kunden-ID-Code gesucht wird, entspricht z. B. die in Fig. 22 gezeigte Bildschirmanzeige der Fensteranzeige zur Eingabe der Startbedingung für eine Suche.

Wie in Fig. 22 gezeigt, werden im Programm des Entwurfsmusters 23 sowohl die Spalte 17A zur Eingabe des Kunden-ID-Codes wie auch Spalte 17B zur Eingabe des Kunden (Initialen) in Kana-Buchstaben angezeigt. Es reicht aus, entweder den ID-Code oder den Kana-Buchstaben in die entsprechende Spalte einzutragen. Angenommen es wird z. B. der ID-Code "0010" in die Eingabespalte 17A eingegeben, werden der ID-Code und der entsprechende Kunde nebeneinander für alle Codes angezeigt, einschließlich und nachfolgend zum ID-Code "0010" (ST63). Wird auf der anderen Seite der Kana-Buchstabe "KA" in der Eingabespalte 17B eingegeben, erscheinen alle Kunden mit den Initialen einschließlich und nachfolgend zu "KA" zusammen mit ihren ID-Codes (ST63).

Ist der Kunden-ID-Code im "Pflegeprogramm für Kundenstammdateien" bereits eingegeben, wird die Verarbeitung des Schritts ST62 übersprungen und die ID-Codes einschließlich und nachfolgend dem eingegebenen ID-Code werden zusammen mit den Kundennamen angezeigt (ST63). Da in jedem Fall eine Vielzahl von Kunden und zugehörige ID-Codes in der Fensteranzeige erscheinen, kann der Anwender den Cursor an die entsprechende Stelle bringen und die "Enter"-Taste drücken (ST63).

Angenommen der gesuchte Kunden-ID-Code sei "0005", darauf kehrt die Programmfolge zum "Pflegeprogramm für Kundenstammdateien" zurück, wobei die ID-Codes gespeichert werden (ST64), daher ist wenn das Aufrufprogramm wieder angezeigt wird "0005" bereits in die Kunden-ID-Code-Spalte eingegeben. Wird

die Funktionstaste F3 oder die Funktionstaste F12 während der Verarbeitung der Schritte ST62 oder ST63 gedrückt, wird das Aufrufprogramm unmittelbar wieder aufgenommen. Daher ist, wenn das Aufrufprogramm angezeigt wird, die Kunden-ID-Code-Spalte leer.

[Entwurfsmuster 24] Programmbaustein WN011

Das Entwurfsmuster 24 (Programmbaustein WN011) stimmt im wesentlichen mit dem Entwurfsmuster 23 (Programmbaustein WN010) überein, der einzige Unterschied besteht darin, daß die Startbedingungen für die Suche in der Verarbeitung des Schritts ST62 nicht in Kana-Buchstaben eingegeben werden können. Das Entwurfsmuster ist daher für Situationen geeignet, in denen eine Suche in einer Stammdatenbank durchgeführt wird, die keine Kana-Buchstaben enthält.

[Entwurfsmuster 25] Programmbaustein WN012

Das Entwurfsmuster 25 (Programmbaustein WN012) ist wie die Entwurfsmuster 23 und 24 eine Verarbeitung, um in einem Fenster der Bildschirmseite in den Daten der Stammdatenbank zu suchen. Die Wirkungsweise ist vergleichbar mit der des Entwurfsmusters 24 (Programmbaustein WN011), dieses Entwurfsmuster ist jedoch dadurch gekennzeichnet, daß in der Verarbeitung des Schritts ST62 unabhängig vom Startcode der Suche ein bestimmter Wert für die Suche festgelegt werden kann. Mit dem festgelegten Wert wird der Wert einer Suchbedingung definiert. Nimmt man eine Suche in einer "Arbeitnehmerdatei", in der die individuellen Daten der Arbeitnehmer enthalten sind, als Beispiel, ist das Entwurfsmuster 25 geeignet, wenn nur im Personalbestand einer bestimmten Abteilung, z. B. der Verkaufsabteilung, gesucht werden soll.

Das obige kann wie folgt genauer erklärt werden: Im Fall des Entwurfsmusters 25 (Programmbaustein WN012) werden, wenn der ID-Code der Verkaufsabteilung als festgelegter Suchwert, sowie der Arbeitnehmercode "0020" als Startcode für die Suche eingegeben werden (ST62), alle Daten der Arbeitnehmer aus der Verkaufsabteilung einschließlich und unterhalb von "0020" zusammen mit den zugehörigen Arbeitnehmercodes am Arbeitsplatz angezeigt (ST63).

[Entwurfsmuster 26] Programmbaustein WN020

Das Entwurfsmuster 26 (Programmbaustein WN020) eignet sich für eine Suche in einer Übergabedatei, sozusagen in den Daten, die mit den Programmen der Entwurfsmuster 1 - 7 gespeichert wurden.

Seine Wirkungsweise ist, wie im Flußdiagramm von Fig. 21 dargestellt, dadurch gekennzeichnet, daß eine Vielzahl von Suchbedingungen im Schritt ST62 eingegeben werden können.

Werden z. B. der ID-Code des Lieferanten "ABC Industrial Co." und das Belegdatum "3.10." bei der Suche in der Spezifikationsdatei 14 in Fig. 8 eingegeben (ST62) werden im Fenster alle Daten angezeigt, die diese Bedingungen erfüllen (ST63).

5. Verarbeitung des Belegdruckersystems
(Entwurfsmuster 27 bis 34)

Entwurfsmuster 27 bis 34 sind Verarbeitungen zum Druck von Belegen oder Kontrollabschnitten. Diese Entwurfsmuster bestehen aus den folgenden Kombina-

tionen der Programmbausteine LT020, LT021, LT030 und LT040.
Entwurfsmuster 27 = LT020
Entwurfsmuster 28 = LT021
Entwurfsmuster 29 = LT010 + LT020
Entwurfsmuster 30 = LT010 + LT021
Entwurfsmuster 31 = LT010 + LT030 + LT020
Entwurfsmuster 32 = LT010 + LT030 + LT021
Entwurfsmuster 33 = LT040 + LT020
Entwurfsmuster 34 = LT040 + LT021

[Entwurfsmuster 27] Programmbaustein LT020

Die Verarbeitung des Entwurfsmusters 27 (Programmbaustein LT020) ist im Flußdiagramm der Fig. 23 skizziert. Dieses Entwurfsmuster ist zur Erstellung eines Ausdrucks des Inhalts einer Datei geeignet, die keine Datenabgrenzung enthält, z. B. eine Stammdatei. Zuerst wird ein Eintrag aus der entsprechenden Datei eingelesen (ST71), und im Fall des EOF (Ende der Datei), stoppt die Verarbeitung (ST72), anderenfalls (falls nicht EOF) wird ein Editervorgang für den Ausdruck durchgeführt (ST73). Beim Editervorgang werden die einzelnen eingelesenen Daten z. B. in einer für den Druck geeigneten Form angeordnet. Es wird daher abgefragt, ob eine Druckseite fertiggestellt wurde oder nicht (ST74), und wenn dies bejaht wird, wird die Titelzeile der nächsten Seite gedruckt (ST75).

Im Fall, daß für die weitere Seite die Titelzeile nicht benötigt wird, weil nur eine Seite folgt, wird eine Seite der editierten Daten ohne Titelzeile gedruckt (ST76), dann wird das nächste Datenäquivalent ausgelesen (ST77) und die Programmfolge kehrt zur Verarbeitung des Schritts ST72 zurück.

[Entwurfsmuster 28] Programmbaustein LT021

Die Verarbeitung des Entwurfsmusters 28 (Programmbaustein LT021) ist in Fig. 24 in einem Flußdiagramm skizziert. Dieses Entwurfsmuster ist dadurch gekennzeichnet, daß ein Verarbeitungsschritt (ST76') zum Ausdruck einer Spezifikationskopfzeile HD zwischen die Verarbeitung des Schritts ST75 und die Verarbeitung des Schritts ST76 eingefügt ist. Mit dem Ausdruck Spezifikationskopfzeile HD ist z. B. der Inhalt der Felder "Belegnummer" 14A1, "Datum" 14A2, "Lieferant" 14A3 und "Einkäufer" 14A4 in der Spezifikationsdatei 14 in Fig. 8 gemeint, bei jedem Ausdruck des Inhalts eines Beleges wird der Inhalt des Kopfzeilenteils 14A mit ausgedruckt (St76').

Im weiteren wird genauer auf das Flußdiagramm aus Fig. 24 eingegangen, wobei auf den in Fig. 25 gezeigten Ausdruck Bezug genommen wird. Zuerst wird ein Eintrag aus der entsprechenden Datei eingelesen (ST71) und, falls EOF (Ende der Datei), endet die Verarbeitung (ST72), sonst (nicht EOF) wird ein Editervorgang für den Ausdruck durchgeführt (ST73). Danach wird abgefragt, ob ein Seitenäquivalent ausgedruckt wurde (ST74), und wenn zugestimmt wird, wird nachdem ein Seitenwechsel durchgeführt wurde die Kopfzeile für eine weitere Seite ausgedruckt (ST75).

Ist der Vorgang eines Seitenwechsels nicht erforderlich, wird abgefragt, ob die Daten eines Belegäquivalents ausgedruckt wurden, und falls zugestimmt wird, werden sowohl der Inhalt des Kopfzeilenteils 14A als auch die Überschrift ausgedruckt (ST76'), wie auch das editierte Datenäquivalent (ST76). Dann wird das nächste Datenäquivalent ausgelesen (ST77) und die Programm-

folge kehrt zur Verarbeitung ST72 zurück.

[Entwurfsmuster 29, 30] Programmbaustein LT010 + LT020, LT021

5 Die Entwurfsmuster 29 und 30 sind so aufgebaut, daß der Verarbeitung der Programmbausteine LT020 und LT021 jeweils die Verarbeitung des Programmbausteins LT010 vorausgeht.
10 Wie in Fig. 26 dargestellt, implementiert der Programmbaustein LT010 jeweils eine Verarbeitungsroutine, mit der der auszudruckende Bereich des Belegs bestimmt wird. Zuerst erscheint am Arbeitsplatz eine Bildschirmmaske zur Eingabe der Ausdruckbedingungen (ST81). Im folgenden wird das Entwurfsmuster 29 an Hand des Inhalts der "Kundenstammdatei" beschrieben. Der Anwender gibt z. B. als Ausdruckbedingung den ID-Code "0001-0010" ein (ST81), worauf eine Bestätigungsaufrufforderung angezeigt wird (ST82). Darauf erscheint eine Meldung "im Druck" (ST83) und die Verarbeitung z. B. eines "Belegausdruck-Programms", welches aus dem Programmbaustein LT020 aufgebaut ist, beginnt abzulaufen (ST84).

Das Entwurfsmuster 30 unterscheidet sich vom gerade beschriebenen Entwurfsmuster 29 nur darin, daß das im Schritt ST84 auferufene Belegdruckprogramm aus dem Programmbaustein LT021 aufgebaut ist.

[Entwurfsmuster 31, 32] Programmbaustein LT010 + LT030 + LT020, LT021

30 Die Entwurfsmuster 31 und 32 sind so aufgebaut, daß die Verarbeitung des Programmbausteins LT030 die Verarbeitung der Entwurfsmuster 29 und 30 mit einschließt. Die Verarbeitung des Programmbausteins LT030 ist, wie in Fig. 27 gezeigt, dadurch gekennzeichnet, daß die Hauptdatei einmalig vollständig eingelesen wird und die Arbeitsdatei aktualisiert wird. Diese Arbeitsdatei ist typischerweise eine Datei für eine Additionsoperation.

Nimmt man den Ausdruck des Inhalts der Spezifikationsdatei 14 aus Fig. 8 als Beispiel, wird ein Datensatzäquivalent aus der Spezifikationsdatei 14 eingelesen (ST91) und falls nicht EOF, die eingelesenen Daten zum Druck editiert (ST93). Danach wird die Additionsoperation der Produkte von "Einzelpreis" multipliziert mit "Einkaufsmenge" durchgeführt (ST94). Anschließend wird das nächste Datenäquivalent eingelesen (ST95) und die Programmfolge kehrt zur Verarbeitung des Schritts ST92 zurück.

Nach der Verarbeitung des Programmbausteins LT030 wird der eigentliche Druck durch die Verarbeitungsroutine des Programmbausteins LT020 oder des Programmbausteins LT021 ausgeführt.

55 [Entwurfsmuster 33, 34] Programmbaustein LT040 + LT020, LT021

60 Die Entwurfsmuster 33 und 34 sind so ausgestaltet, daß die Verarbeitung des Programmbausteins LT040 vor der Verarbeitung der Entwurfsmuster 27 und 28 ausgeführt wird. Wie in Fig. 28 gezeigt, werden die Bedingungen für den Ausdruck von Daten durch den Ablauf des Programmbausteins LT040, wie auch des Programmbausteins LT010 bestimmt. Der Unterschied zwischen den Programmbausteinen LT010 und LT040 besteht darin, daß beim Programmbaustein LT040 eine Liste der Daten, welche die Bedingung erfüllen, ange-

zeigt wird. Es wird im folgenden das Flußdiagramm aus Fig. 28 erklärt, wobei der Ausdruck des Inhalts der in Fig. 5 gezeigten Übergabedateien 12, 13 als Beispiel herangezogen wird.

Zuerst erscheint am Arbeitsplatz eine Anzeige zur Eingabe der Druckbedingungen. Es wird jetzt angenommen, daß der Anwender als Bedingung "1.10. bis 5.10.1994" eingibt (ST101).

Danach werden, wie in Fig. 14 dargestellt, alle Daten angezeigt, die diese Bedingungen erfüllen (ST102). Zur selben Zeit erscheint auch die Spalte für die Eingabe des Funktionscodes auf dem Bildschirm des Arbeitsplatzes, und der Anwender gibt daher den entsprechenden Funktionscode ein (ST103).

Darauf wird eine Bestätigung der Eingabe angefordert (ST104). Es erscheint dann eine Anzeige "im Druck" (ST105), und das Formblattprogramm wird aufgerufen (ST106). Das Formblattprogramm wurde entweder mit dem Programmbaustein LT020 oder dem Programmbaustein LT021 erstellt, ob das Entwurfsmuster 33 oder das Entwurfsmuster 34 ausgewählt wird, hängt davon ab, welcher der Programmbausteine LT020 und LT021 verwendet wurde.

6. Verarbeitung des Systems zur Aktualisierung von Stapeln (Entwurfsmuster 35, 36)

[Entwurfsmuster 35] Programmbaustein BU020

Das Entwurfsmuster 35 (Programmbaustein BU020) ist eine Verarbeitungsroutine mit der eine Datei ohne einen Dialog am Arbeitsplatz aktualisiert werden kann. Typischerweise entspricht dieser Fall einer Verarbeitung, in der Daten aus einer Übergabedatei zur Aktualisierung des Inhalts einer Stammdatei Eintragsäquivalent für Eintragsäquivalent ausgelesen werden. Des weiteren entspricht die Anlage einer temporären Datei (Arbeitsdatei), in der das Abbild eines Ausdrucks eines Beleges (z. B. Fig. 25) abgespeichert ist, ehe die in der Übergabedatei angesammelten Daten ausgedruckt werden, ebenfalls dieser Verarbeitung.

Die obige Verarbeitung wird nun in Bezugnahme auf das Flußdiagramm in Fig. 29 erklärt. Zuerst wird der Anfangseintrag einer Übergabedatei eingelesen (ST111) und (falls nicht EOF), eine Editierung der Spezifikation durchgeführt (ST113). Unter Editierung der Spezifikation wird eine Verarbeitung zur Editierung eines einzelnen Eintragsäquivalents von eingelesenen Spezifikationsdaten im Format der Stammdatei, der temporären Datei o. ä. verstanden, das aktualisiert werden muß.

Danach wird basierend auf dem einzelnen Datensatz der editierten Daten die Stammdatei, die temporäre Datei o. a. aktualisiert (ST114). Dann wird das nächste Eintragsäquivalent aus der Übergabedatei eingelesen (ST115), und die Programmabfolge kehrt zur Verarbeitung des Schritts ST112 zurück.

[Entwurfsmuster 36] Programmbaustein BU010 + BU020

Entwurfsmuster 36 ist eine Routine, in der die Aktualisierungsbedingung vor der Verarbeitung des Entwurfsmusters 35 eingegeben wird.

Wie in Fig. 30 gezeigt, erscheint zuerst eine Anzeige zur Eingabe der Aktualisierungsbedingung auf dem Bildschirm (ST121). Hier wird als Beispiel eine Aktualisierung einer Stammdatei gezeigt, mit der eine Rech-

nung auf Basis der Daten vom 21. September bis 20. Oktober erstellt wird, die in der Übergabedatei aus Fig. 11 gespeichert sind. Der Anwender gibt z. B. die Aktualisierungsbedingung "21.9. bis 20.10." am Arbeitsplatz ein (ST121).

Darauf wird zur Bestätigung der Daten aufgefordert (ST122). Sowie der Anwender die "Enter"-Taste drückt, erscheint eine Anzeige "Aktualisieren" (ST123), und das aus dem Programmbaustein BU020 aufgebaute Aktualisierungsprogramm wird aufgerufen (ST124). Daraufhin wird, entsprechend dem Ablauf des Entwurfsmusters 35, nur der entsprechende Eintrag der Übergabedatei aus Fig. 11 aufgerufen, und der Inhalt der Stammdatei aktualisiert.

Die verschiedenen Entwurfsmuster sind bereits eingehend beschrieben worden. Es wird jetzt genauer auf den Ablauf der automatischen Erstellung eines Programms im Dialog eingegangen. Als Beispiel wird die Fertigstellung eines "Pflegeprogramms für Arbeitnehmerstammdateien" beschrieben. Das "Pflegeprogramms für Arbeitnehmerstammdateien" ist ein Anwendungsprogramm, das mit dem Computersystem aus Fig. 1 betrieben wird. Es ist ein Programm, mit dem auf die im Plattspeicher 3 gespeicherte "Arbeitnehmerstammdatei" von den Arbeitsplätzen WS1 bis WSN aus zugegriffen wird, um die Daten der einzelnen Arbeitnehmer zu registrieren, zu überarbeiten oder zu löschen.

Die erfundungsgemäße automatische Erstellung eines Programms ist unterteilt in die Verarbeitungsschritte (1A) physikalischer Entwurf einer Datei, (1B) Entwurf der Bildschirmmaskendatei, (1C) Entwurf der Formblattdatei (2), Registrierung der Einheiteninformation und (3) detaillierter Programmewurf. Da im Fall des "Pflegeprogramms für Arbeitnehmerstammdateien" kein Vorgang zum Ausdruck von Belegen benötigt wird, umfaßt die automatische Generierung des "Programms zur Pflege von Arbeitnehmerstammdateien" nur die Erstellung der physikalischen Datei (Fig. 31), den Entwurf der Bildschirmmaskendatei (Fig. 34), die Eingabe der benötigten Einheiten (Fig. 38) und die detaillierte Programmerstellung (Fig. 39).

(1A) Entwurf einer physikalischen Datei (Fig. 31)

Der Computer wird in den Funktionsmodus der Erstellung einer physikalischen Datei gesetzt und der Entwurf der physikalischen Datei im Dialog ausgeführt. Da die Aufgabe in diesem Beispiel darin besteht, eine "Arbeitnehmerstammdatei" zu entwerfen, wird abschnittsweise festgelegt, welche Felder für einen Eintrag zur Verfügung gestellt werden müssen, wie viele Bytes für den jedem Feld zugewiesenen Bereich benutzt werden sollen, und welches Feld für die Daten ausgewählt werden soll, die als Suchbegriff bei einer Suche eingesetzt werden sollen usw. (ST131).

Nach Vollendung der Eingabe aller Informationen kann ein Entwurfsdokument der Datenbasis ausgedruckt werden (ST132).

Fig. 32 zeigt ein Entwurfsdokument einer Datenbasis. Da der "Feldname" 2A, die "Feld-ID" 20C, der "Feldtyp" 20B, die "Feldstellenzahl" 20D, der "Datentyp" 20E etc. entsprechend einem vorbestimmten Format angezeigt werden, kann der Inhalt des Entwurfs mit einem flüchtigen Blick überprüft werden. Des weiteren kann, solange dieses Entwurfsdokument der Datenbasis gespeichert ist, jeder leicht den Inhalt der "Arbeitnehmerstammdatei" erfassen.

Nach Vollendung der Erstellung der physikalischen

Datei im Dialog werden automatisch Quellcodes der physikalischen Datei erzeugt (ST133). Dann kann die Quelliste der physikalischen Datei ausgedruckt werden (ST134). Fig. 33 zeigt die Quelliste. Auf diese Weise wird ein Ausdruck der Quelliste erhalten, der in vollkommener Übereinstimmung mit dem Entwurfsdokument aus Fig. 32 steht.

Dann werden die Quellcodes kompiliert und ein Bereich für die "Arbeitnehmerstammdaten" im Plattspeicher 3 gesichert (ST135). In diesem Beispiel lautet der ID der "Arbeitnehmerstammdaten" "RCMEMPP", wobei die ID-Felder "EMPCDE" bis "EMPODT" definiert werden.

(1B) Entwurf der Bildschirmmaskendatei (Fig. 34)

Als nächstes wird, nachdem der Computer in den Funktionsmodus der Bildschirmmaskendateierstellung gesetzt wurde, der Entwurf der Bildschirmmaskendatei im Dialog durchgeführt (ST141). Beim Entwurf der Bildschirmmaskendatei muß zunächst festgelegt werden, welcher Programmbaustein eingesetzt wird. In diesem Beispiel wird das "Pflegeprogramms für Arbeitnehmerstammdaten" unter Gebrauch des Programmbausteins MM011 (Entwurfsmuster 16) erstellt.

Wie in Fig. 19 dargestellt, benötigt der Programmbaustein MM011 (Entwurfsmuster 16) eine Bildschirmmaske zur Eingabe der Suchbegriffe (ST52), eine Bildschirmmaske zur Eingabe der Spezifikationsdatei (ST53) und eine Bildschirmmaske zur Anzeige einer Bestätigungsaufforderung mit einer damit einhergehenden Anzeige der Spezifikationsdaten (ST54). Diese werden schrittweise erstellt.

Nach Vervollständigung aller Entwürfe der Bildschirmmasken durch Verarbeitung des Schritts ST141 kann ein Dokument der Bildschirmmaskendatei ausgedruckt werden (ST142).

Dieses Dokument der Bildschirmmaskendateierstellung legt detaillierte Informationen über alle Bildschirmanzeigen (ST52, ST53, ST54) dar, der Inhalt der Erstellung der Bildschirmmasken kann aus diesem Erstellungsdokument der Bildschirmmaskendatei genau überprüft werden. Fig. 35 und 36 zeigen "Bildgestaltungen" als Teile des Entwurfsdokuments einer Bildschirmmaskendatei. Es ist offensichtlich, daß im Schritt ST52 in Fig. 19, die Bildschirmanzeige am Arbeitsplatz der Darstellung in Fig. 35 (FMT01 + FMT91) entspricht und daß in den Schritten 53 und 54 in Fig. 19 die Bildschirmanzeige der Darstellung in Fig. 36 entspricht (FMT02 + FMT01 + FMT91).

Nach Vollendung des Entwurfs der Bildschirmmaskendatei werden automatisch Quellcodes für die Bildschirmmaskendatei erzeugt (ST143). Dann kann eine Quelliste der physikalischen Datei ausgedruckt werden (ST144). Fig. 37 zeigt einen Teil der Bildschirmmaskendateiquelliste. Im besonderen zeigen die 17. bis 52. Zeile eine Quelliste, die sich auf die Bildschirmmaske (FMT01) zur Eingabe der Suchbegriffe in Schritt ST52 bezieht (Fig. 19).

Danach werden die Quellcodes kompiliert und die Maschinencodes der Bildschirmmaskendatei erzeugt (ST145). Der Name dieser Bildschirmmaskendatei ist "Pflege der Arbeitnehmerstammdaten" und die 1D der Datei ist "RC00D001".

(2) Registrierung der Einheiteninformation (Fig. 38)

Nun sollte der Computer in den Modus der Registrie-

rung der Einheiteninformation versetzt werden. Im Registrationsmodus für die Einheiteninformation werden im Dialog die Grundzüge für das "Programm zur Pflege von Arbeitnehmerstammdaten" festgelegt.

- 5 Wie in Fig. 38 gezeigt, muß zuerst der Name des zu erstellenden Programms sowie der zu benutzende Programmbaustein festgelegt werden (ST151). Hier wird, wie oben erwähnt, unter Verwendung des Programmbausteins MM011 ein "Pflege von Arbeitnehmerstammdaten" genanntes Programm erstellt. Diese Information kann daher einfach vom Arbeitsplatz aus eingegeben werden (ST151).

Danach werden, sollte das zu erstellende Anwendungsprogramm mit einem anderen Programm verbunden werden, die Parameter festgelegt, die vom anderen Programm erhalten werden sollen (ST152). In diesem Beispiel wird kein Parameter aus einem anderen Programm erhalten.

- 15 Dann werden die Dateien, die vom "Programm zur Pflege von Arbeitnehmerstammdaten" verwendet werden sollen, festgelegt (ST153). In diesem Beispiel wird die "Arbeitnehmerstammdaten (RCMEMPT)" als physikalische Datei benutzt, und die Datei "Pflege von Arbeitnehmerstammdaten (RC00D001)" wird als Bildschirmmaskendatei bestimmt.

Zuletzt wird ein externes Programm bestimmt, welches während der Ausführung des Programms "Pflege von Arbeitnehmerstammdaten" aufgerufen werden soll (ST154). In diesem Beispiel wird eine Verarbeitung zur Code suche mit der Funktionstaste F4 (Fig. 19) benutzt, daher wird ein externes Programm zur Codesuche ausgewählt. Es sei angenommen, daß das Programm zur Code suche bereits unter Verwendung irgendeines der Programmbausteine WN010, WN011 und WN012 erstellt wurde.

(4) Detaillierte Programmerstellung (Fig. 39)

Nach Festlegung der benötigten Einheiten wird, wie in Fig. 39 gezeigt, ein detailliertes Programm erstellt. Hier wird im Dialog detailliert die Information eingegeben, die mit den individuellen Verarbeitungen variiert (ST161). Da der Programmbaustein MN011 in einen Basisteil A und einen individuellen Teil B aufgeteilt ist (Fig. 2), um entsprechend dem Vorgehen in Fig. 19 eine Vielzahl von Verarbeitungen durchführen zu können, wird der wesentliche Teil des "Programms zur Pflege von Arbeitnehmerstammdaten" in seinen Einzelheiten im Schritt ST161 festgelegt.

- 50 Nach Vollendung der obigen Verarbeitungen kann ein Dokument des Programmentwurfs ausgedruckt werden (ST162). Im in Fig. 1 gezeigten automatischen Programmgenerator ist im Plattspeicher 3 für jeden Programmbaustein 1 bis 36 ein Entwurfsdokument in halbfertigem Zustand gespeichert. Daher kann nach Fertigstellung der obigen detaillierten Programmerstellung (ST161) ein Erstellungsdokument das in genauer Übereinstimmung mit dem entsprechenden Programm steht, als Ausdruck zur Verfügung gestellt werden.

Des weiteren werden nach Vollendung der detaillierten Programmerstellung die Quellcodes des "Programms zur Pflege von Arbeitnehmerstammdaten" automatisch erstellt (ST163), in dem die bisher eingegebenen Informationen mit dem Quellprogramm des Programmbausteins MN011, welches vorher ausgewählt worden war, kombiniert werden. Die Quellcodes werden kompiliert und ein Maschinenprogramm erzeugt (ST165). Des weiteren kann eine Programmquelliste

ausgedruckt werden (ST164).

Fig. 40 zeigt einen Teil des Quellprogramms des Programmbausteins MN011 und Fig. 41 zeigt einen Ausschnitt aus den Quellcodes des aus gedruckten "Programm zur Pflege einer Arbeitnehmerstammdatei". Weist der Quellcode des "Programm zur Pflege einer Arbeitnehmerstammdatei" Spalten auf, die mit den Zeichen "/" oder "//" ausgefüllt sind, sind die Zeilen mit dem Zeichen "/" eine direkte Kopie aus dem Quellprogramm des Programmbausteins MM011, eine Zeile mit dem Zeichen "/" oder dem Leerzeichen stellt eine Zeile aus der automatischen Generierung dar, die sich auf die im Dialog eingegebene individuelle Information bezieht. Die Zeile, welche in Fig. 40 mit "PLIST COMMON" bezeichnet ist, stellt eine Einfügung der Zeilen 136 bis 143 aus Fig. 41 dar, und die Zeilen, welche mit "KLIST COMMON" und "MAKING NG" bezeichnet sind, sind Einfügungen aus den Zeilen 150 bis 155 in Fig. 41. Des weiteren entspricht die mit "COPYRIGHT" bezeichnete Zeile in Fig. 40 der Zeile 159 in Fig. 41.

Bei dem automatischen Programmgenerator aus Fig. 1 werden alle Anwendungsprogramme durch Kombination von Programmbausteinen erzeugt, die Quellprogramme in halbfertigem Zustand darstellen. Diese Programmbausteine sind Programme hoher Qualität, die von fachmännischen Programmierern erstellt wurden, die in der speziellen Programmiersprache ausgebildet sind, wobei die Programmerstellung im Dialog erfolgt. Daher kann selbst ein Novize eine automatische Erstellung von Anwendungsprogrammen gleichbleibender Qualität leicht und schnell durchführen.

Des Weiteren kann, da ein komplettes Entwurfsdokument, welches in vollkommener Übereinstimmung mit dem vollen Anwendungsprogramm steht, als Ausdruck zugänglich gemacht wird, jedermann genau den Inhalt der Verfahren des Programms erfassen. Daher kann, selbst wenn ein Wechsel im Betrieb oder der Dienstleistung der Organisation erfolgt, jedermann leicht das Anwendungsprogramm überarbeiten oder erweitern.

Patentansprüche

1. Automatischer Programmgenerator bestehend aus
 - einem Bausteinspeichermittel zur Speicherung von Programmbausteinen, die Quellprogramme in halbfertiggestelltem Zustand darstellen,
 - einem Entwurfsdokumentspeichermittel zur Speicherung von Entwurfsdokumenten in halbfertiggestelltem Zustand, die den Inhalt der Verarbeitung der Programmbausteine beschreiben,
 - einem Eingabemittel für individuelle Informationen, dessen Aufgabe darin besteht, während der Erstellung von Anwendungsprogrammen im Dialog individuelle Informationen, die mit den unterschiedlichen Anwendungsprogramm variieren, zu empfangen,
 - einem Mittel zur Fertigstellung von Programmen, welches die Programmbausteine auf Grundlage der erhaltenen individuellen Information editiert und Quellcodes für die entsprechenden Anwendungsprogramme erzeugt,
 - einem Mittel zur Fertigstellung von Entwurfsdokumenten, welches die in halbfertiggestelltem Zustand vorliegenden Entwurfsdo-

kumente auf Grundlage der erhaltenen individuellen Informationen editiert und die den jeweiligen Anwendungsprogrammen entsprechenden Entwurfsdokumente fertigstellt und — einem Ausgabemittel, das auf eine entsprechende Handlung hin eine Liste der Quellcodes und/oder des Entwurfsdokuments jedes vollen Anwendungsprogramms ausdrückt.

2. Automatischer Programmgenerator nach Anspruch 1, dadurch gekennzeichnet, daß das Bausteinspeichermittel jeweils im halbfertigen Zustand zumindest eine erste Gruppe Programme speichert, zum Abspeichern von Daten in einer Übergabedatei, eine zweite Gruppe Programme, zum Suchen von Daten in einer Datei und zum Anzeigen auf einem Bildschirm am Arbeitsplatz, eine dritte Gruppe Programme, zum Abspeichern von Daten in einer Stammdatei, eine vierte Gruppe Programme, zum Suchen von Daten in einer Datei und zum Anzeigen der Daten in einem Fenster auf dem Bildschirm am Arbeitsplatz, eine fünfte Gruppe Programme, zum Ausdrucken des Inhalts einer Datei auf einem Formblatt und eine sechste Gruppe Programme, zum Aktualisieren des Inhalts einer Stammdatei oder einer temporären Datei unter Verwendung der Daten aus einer Übergabedatei.
3. Automatischer Programmgenerator nach Anspruch 1 oder 2, dadurch gekennzeichnet, daß das Speichermittel für Entwurfsdokumente zumindest ein Entwurfsdokument für eine physikalische Datei, ein Entwurfsdokument für eine Bildschirmanzeigedatei, ein Entwurfsdokument für eine Formblattdatei und ein Entwurfsdokument für die Programmerstellung jeweils in halbfertiggestelltem Zustand enthält.
4. Automatischer Programmgenerator nach Anspruch 1, 2 oder 3, dadurch gekennzeichnet, daß die Programme der ersten Gruppe eingeteilt werden in Programme zur Verarbeitung von belegartigen Daten, die eine Belegnummer und eine Zeilennummer enthalten und in Programme zur Verarbeitung von editierbaren Daten, die weder eine Belegnummer noch eine Zeilennummer enthalten.
5. Automatischer Programmgenerator nach Anspruch 1, 2, 3 oder 4, dadurch gekennzeichnet, daß die Programme der zweiten Gruppe eingeteilt sind in Programme, die die Aufgabe haben, den Anwender auf zu fordern, die Bedingungen für die gesuchten Daten einzugeben und eine Liste all der Daten anzugeben, die der Bedingung genügen, und in Programme, die nicht die Aufgabe haben, eine Liste anzugeben.
6. Automatischer Programmgenerator nach Anspruch 1, 2, 3, 4 oder 5, dadurch gekennzeichnet, daß die Programme der dritten Gruppe in eine Anzahl von Programmen eingeteilt werden, die auf einer quantitativen Beziehung zwischen einem Äquivalent von Daten und der Datenmenge beruhen, die auf einer einzelnen Bildschirmseite am Arbeitsplatz angezeigt werden kann.
7. Automatischer Programmgenerator nach Anspruch 1, 2, 3, 4, 5 oder 6, dadurch gekennzeichnet, daß die Programme der vierten Gruppen eingeteilt werden in Programme, um eine Stammdatei zu durchsuchen und Programme, um eine Übergabedatei zu durchsuchen.
8. Automatischer Programmgenerator nach An-

spruch 1, 2, 3, 4, 5, 6 oder 7, dadurch gekennzeichnet,
daß die Programme der sechsten Gruppe eingeteilt
werden in Programme, welche die Aufgabe haben,
den Anwender auf zufordern, die Bedingung für die
auszudruckenden Daten einzugeben und eine Liste
der Daten anzuzeigen, die dieser Bedingung genü-
gen und in Programme, die nicht die Aufgabe ha-
ben, eine Liste anzuzeigen.

5

9. Automatischer Programmgenerator nach einem
der vorstehenden Ansprüche, dadurch gekenn- 10
zeichnet, daß die Programmiersprache des Anwen-
dungsprogramms RPG-Sprache, C-Sprache oder
COBOL-Sprache ist.

Hierzu 40 Seite(n) Zeichnungen

15

20

25

30

35

40

45

50

55

60

65

- Leerseite -

THIS PAGE BLANK (USPTO)

Fig. 1

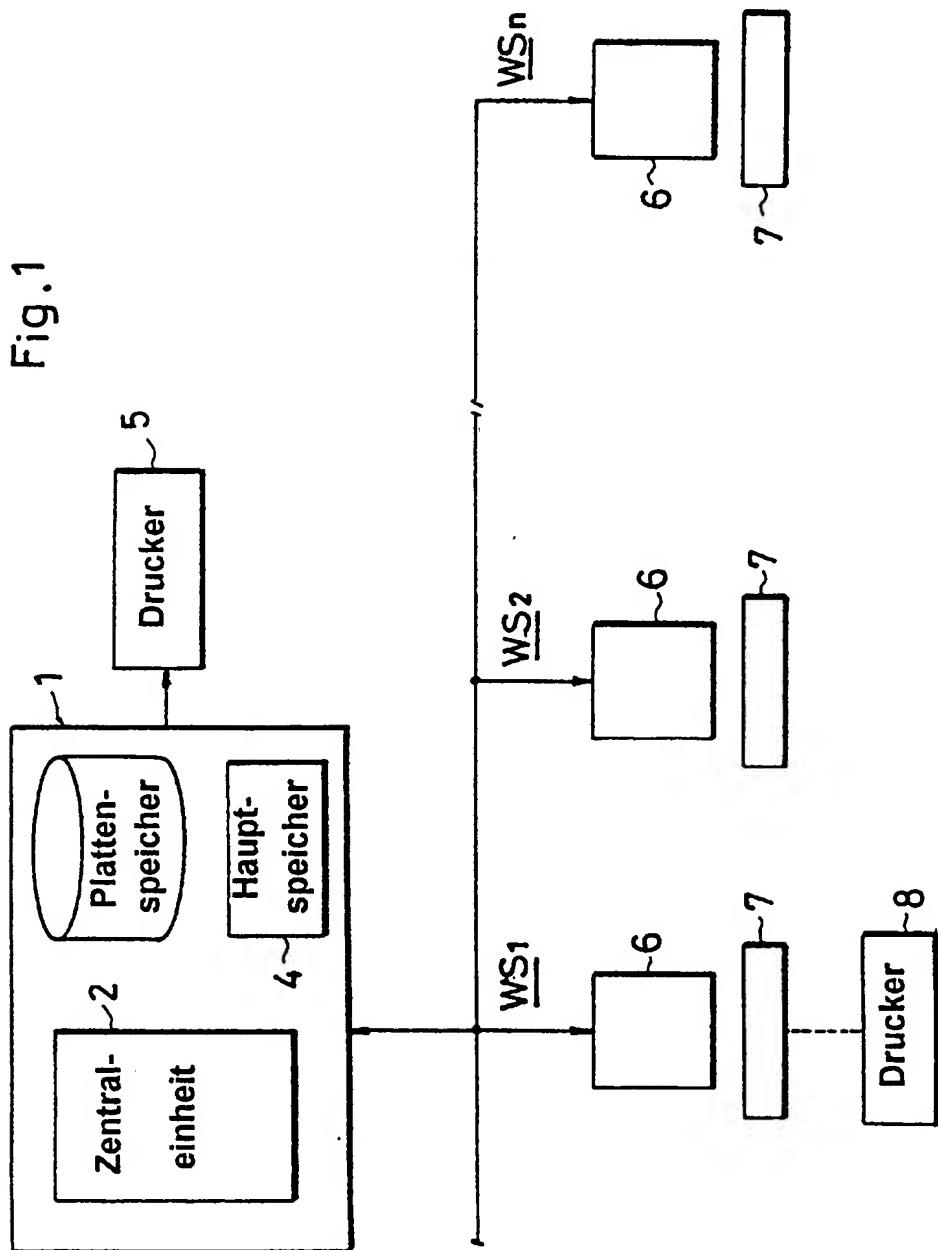


Fig. 2

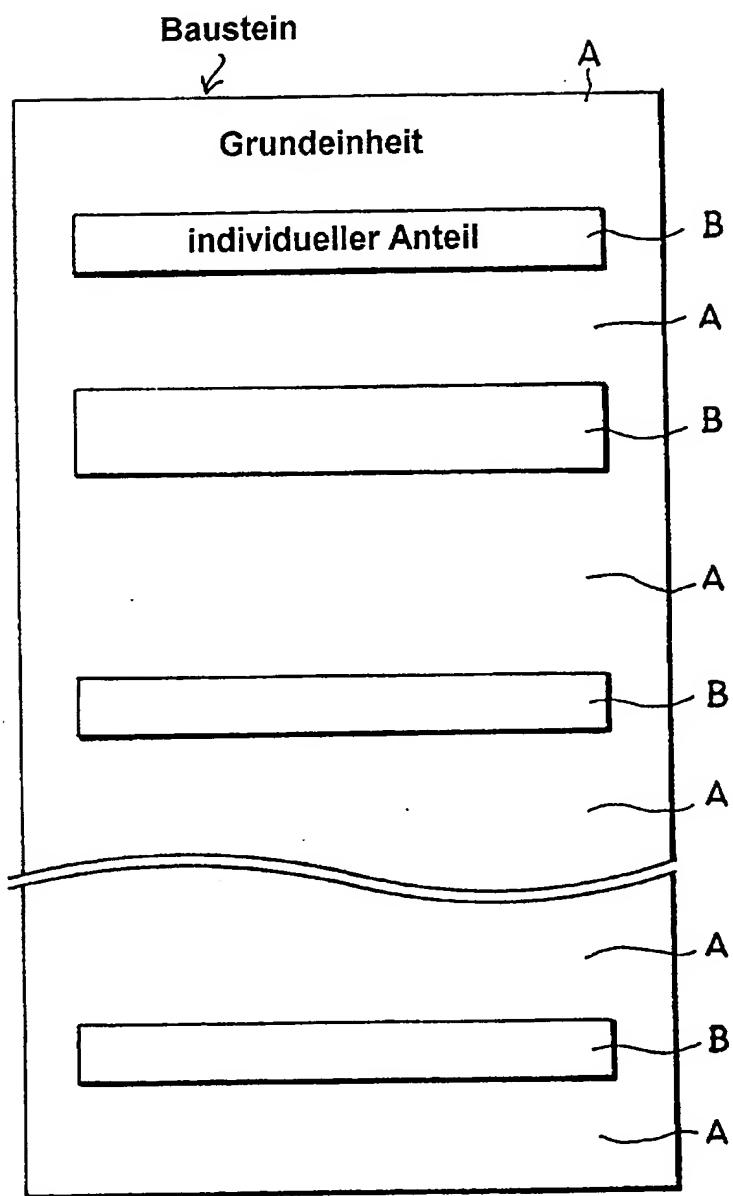


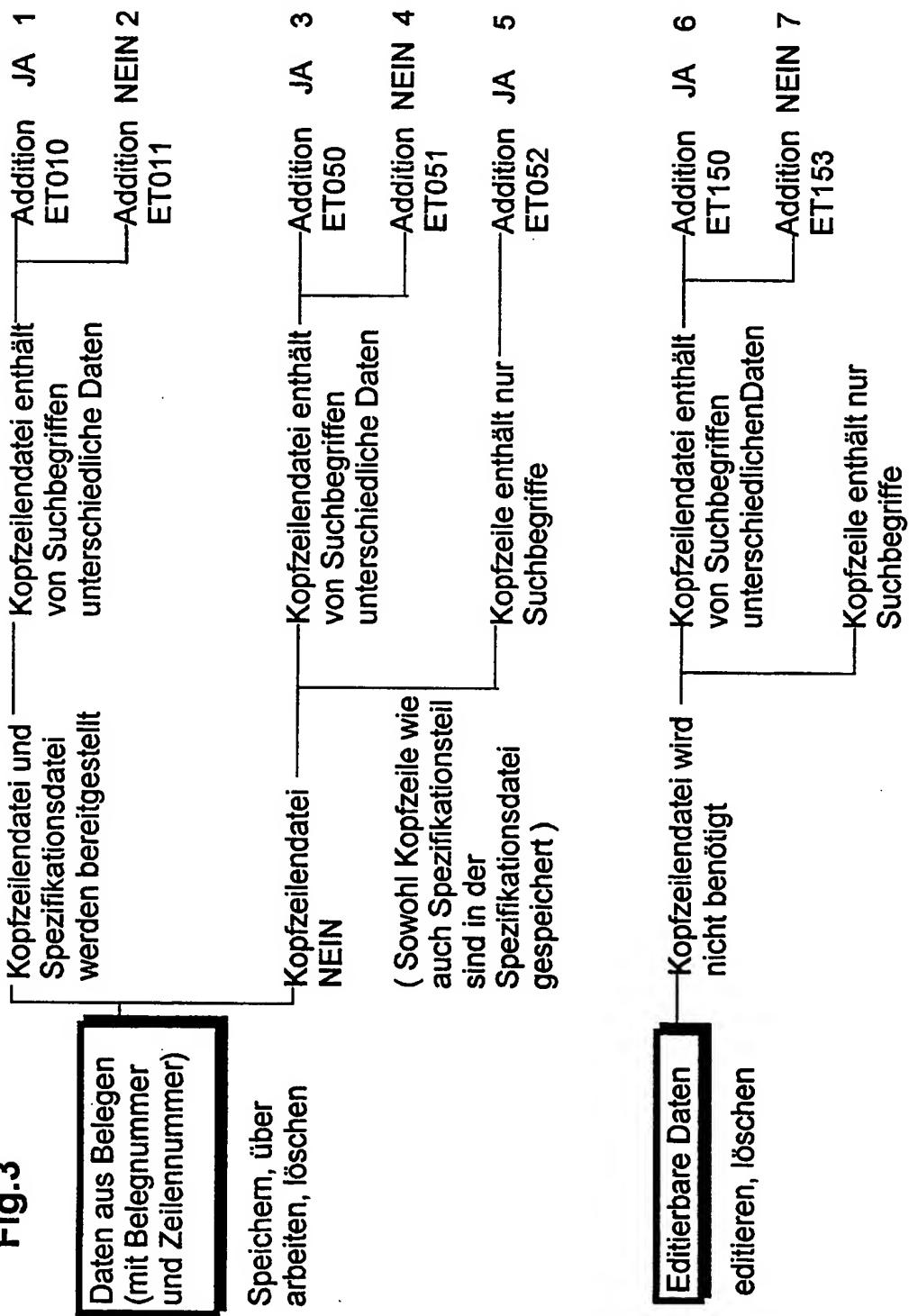
Fig.3

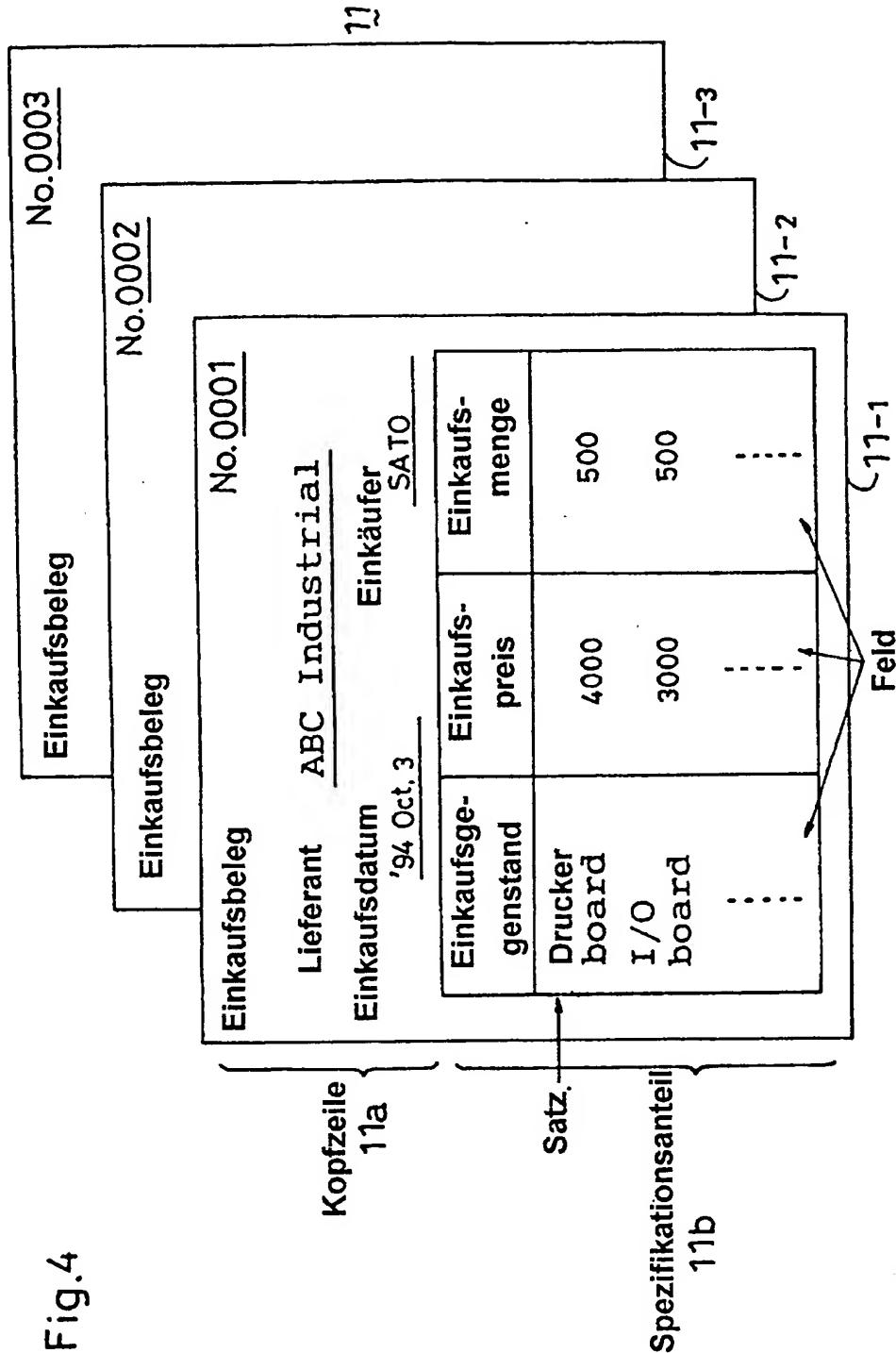
Fig.4

Fig. 5A

[Kopfzeilendatei]

Beleg	No.	Datum	Lieferant	Einkäufer	Summe
0001		Oct.3	ABC Industrial	Sato	
0002			BTF Electronics	Nakamura	

12a~ 12b 12c 12d 12e

K=Suchbegriff

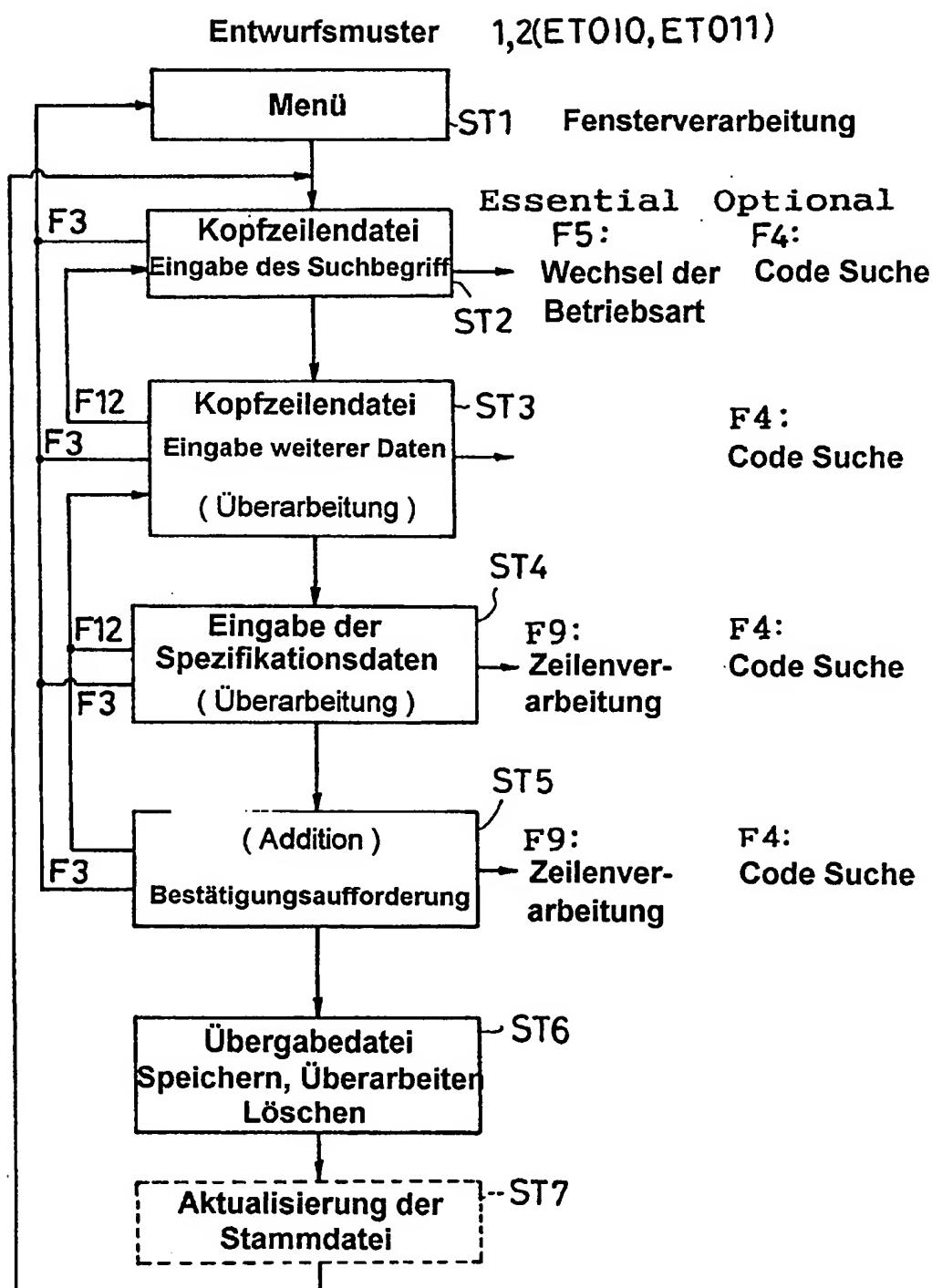
[Spezifikationsdatei]

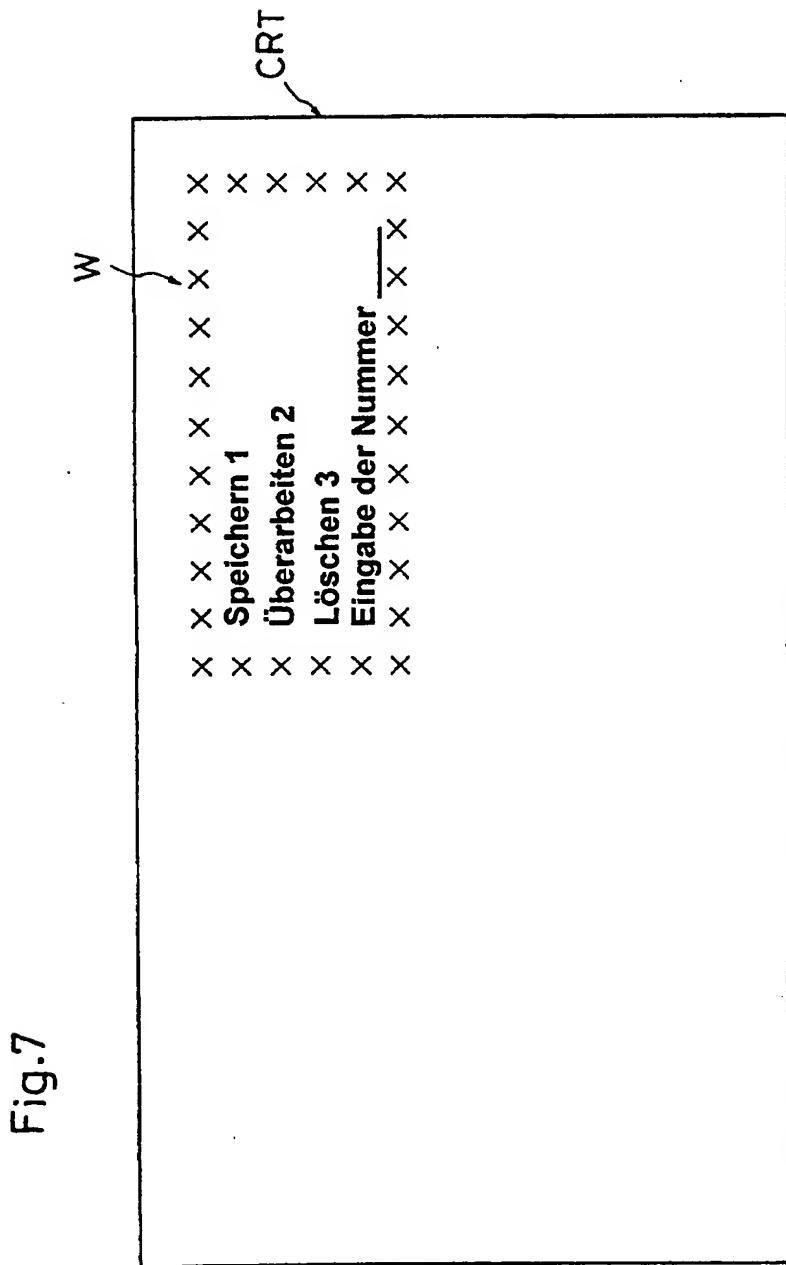
Fig. 5B

Beleg		Datum	Lieferant	Zeile Nr.	Einkaufsge- genstand	Einzelpreis	Einkaufs- menge
0 0 0 1	Oct.3	ABC Industrial		1	Printer board	4 000	500
0 0 0 1	Oct.3	ABC Industrial		2	I/O board	3000	500
				3			
0 0 0 2	Oct.3	BTF Electronics		1	Bolt	10	10000
				2	Nut	5	20000
				3			
0 0 0 3	Oct.3	CTT		1	A1 case	150	500

K=Suchbegriff

Fig.6





[Spezifikationsdatei]

Fig.8

Kopfzeile				Spezifikationsteil				14B				
14A1		14A2		14A3		14A4		14B1	14B2	14B3	14B4	14B5
Beleg Nr.	Datum	Liefer- ant	Einkäufer	Zeile Nr.	Einkaufsge- genstand		Preis		Einkaufs- menge		Gesamt	
0 0 0 1	Oct.3	ABC	Sato	1	Drucker board		4000		500			
	1	Oct.3	ABC	2	I/O board				3000			
				3					500			
0 0 0 2	Oct.3	BTF	Tanaka	1	Bolt							
				2	Nut							
				3								
0 0 0 3	Oct.3	CTT	Suzuki		A1 case							

K=Suchbegriff

Fig.9

Entwurfsmuster 3,4,5 (ET050, ET051, ET152)

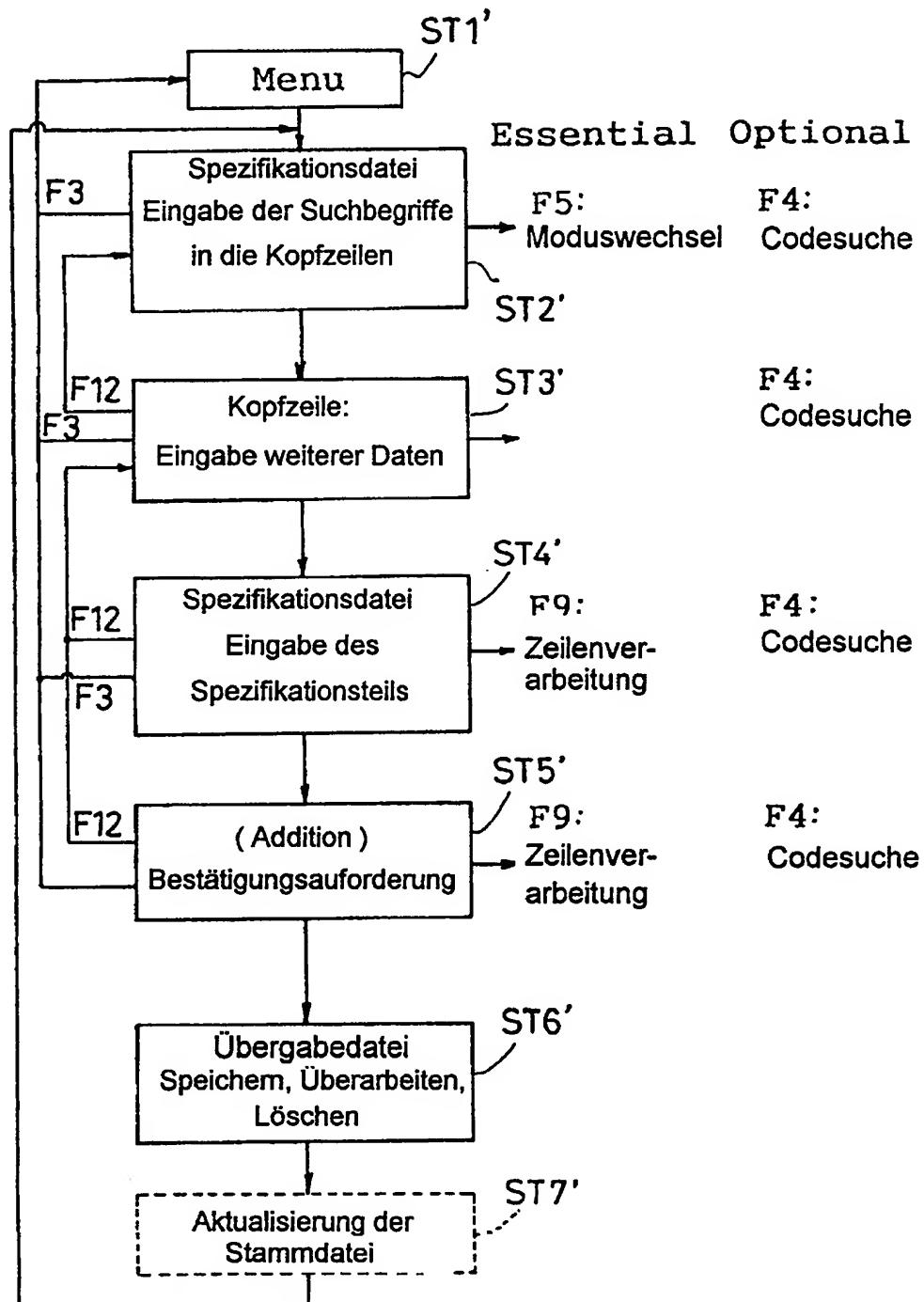


Fig.10

Nakao

SEP.21- SEP.28

Datum	Kunde	Dienstleistung	Zeit	Preis
SEP.21	RCC	Inspektion	1	2000
SEP.21	NTT	Reparatur	1	10000
	JR	Instal- lation	4	5000
SEP.22	JR	Instal- lation	8	5000
SEP.22	JR	Instal- lation	8	5000

Sato

SEP.21- SEP.28

Datum	Kunde	Dienstleistung	Zeit	Preis
SEP.21	JT	Reparatur	2	10000
	ST	Inspektion	4	2000

Fig.

K=Suchbegriff

Fig.12

Musterentwurf 6,7(ET150, ET153)

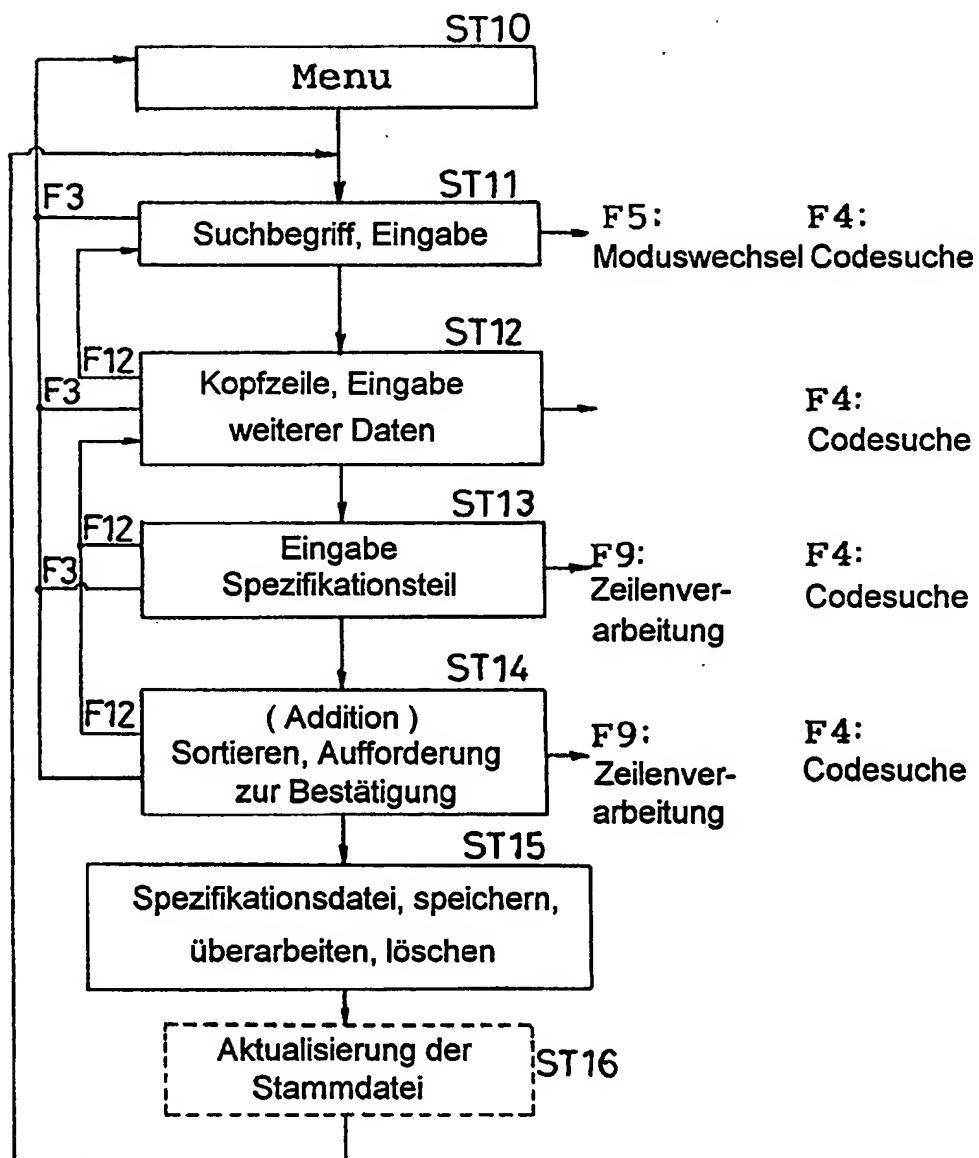


Fig.13

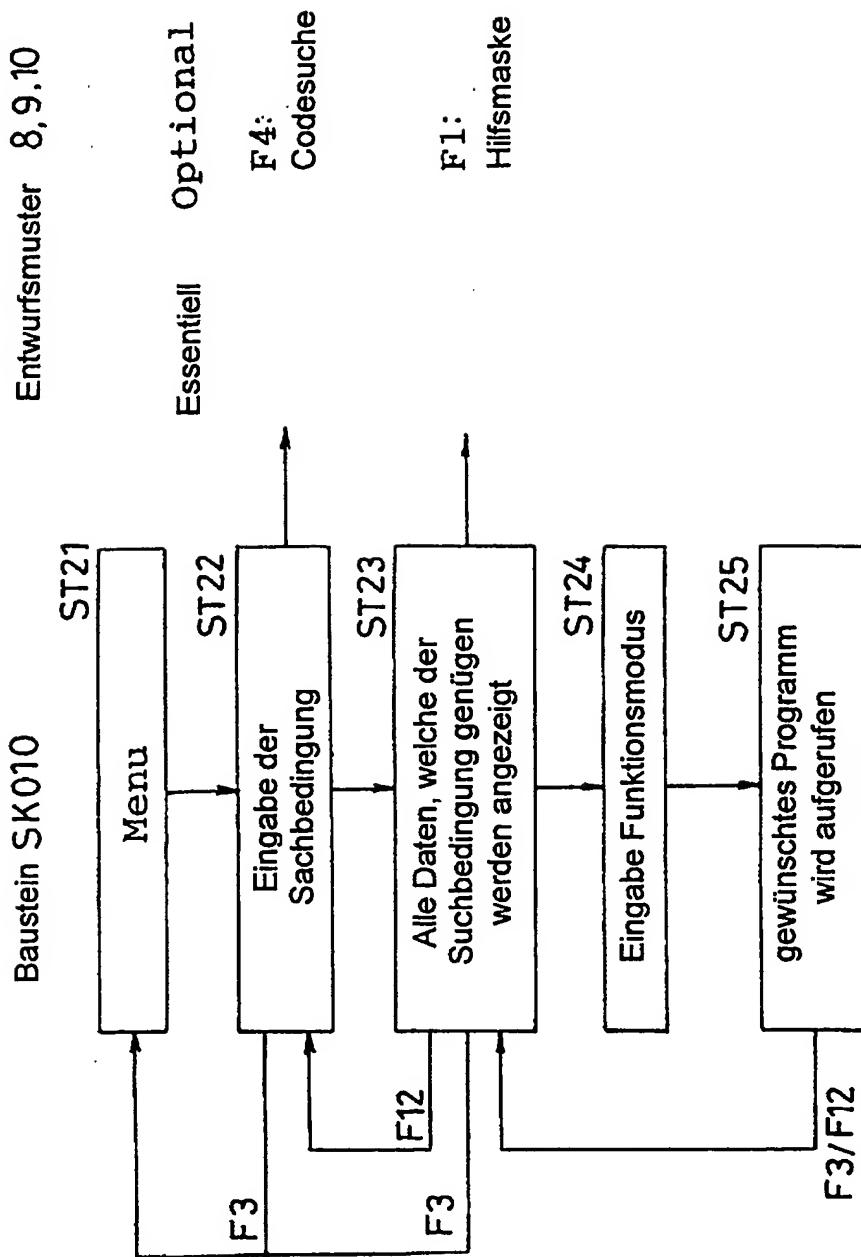


Fig. 14

Funktions- code	Beleg Nr.	Datum	Lieferant	Einkäufer
[]	0001	Oct. 3	ABC Industrial BTTF	Sato
[]	0002	.	Electronics	Nakamura
[]	:	.	:	..
[]	:	.	:	..

(1 PRG1
2 PRG2)

Fig.15

Funktions- code	Kunden- code	Firmenname
[]	0020	- - - - -
[]	0021	- - - - -
[]	0022	- - - - -
[]	0030	[]
		$\begin{pmatrix} 1 & \text{PRG1} \\ 2 & \text{PRG2} \\ 3 & \text{PRG3} \end{pmatrix}$

Fig.16

Entwurfsmuster 11 (SK041)

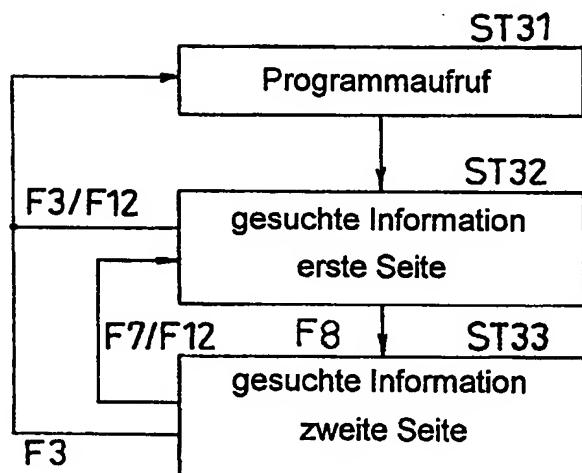


Fig.17

Entwurfsmuster 12-14(SK050,SK051,SK60)

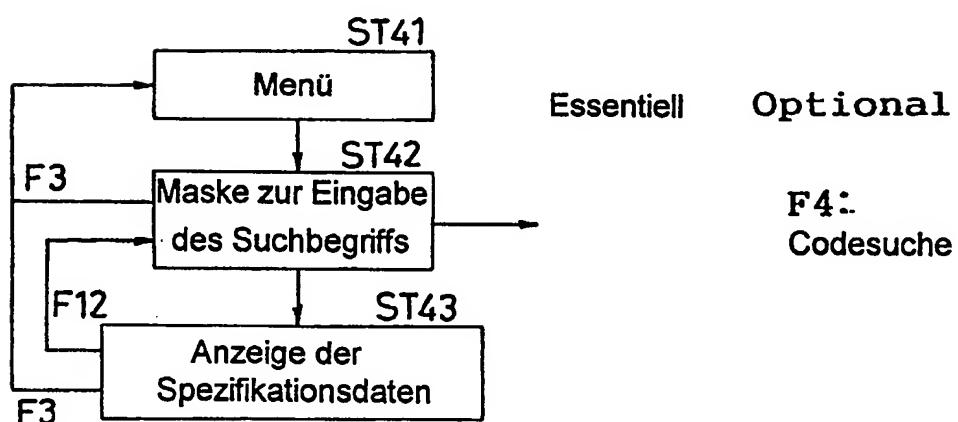


Fig. 18

K=Suchbegriff

Fig.19

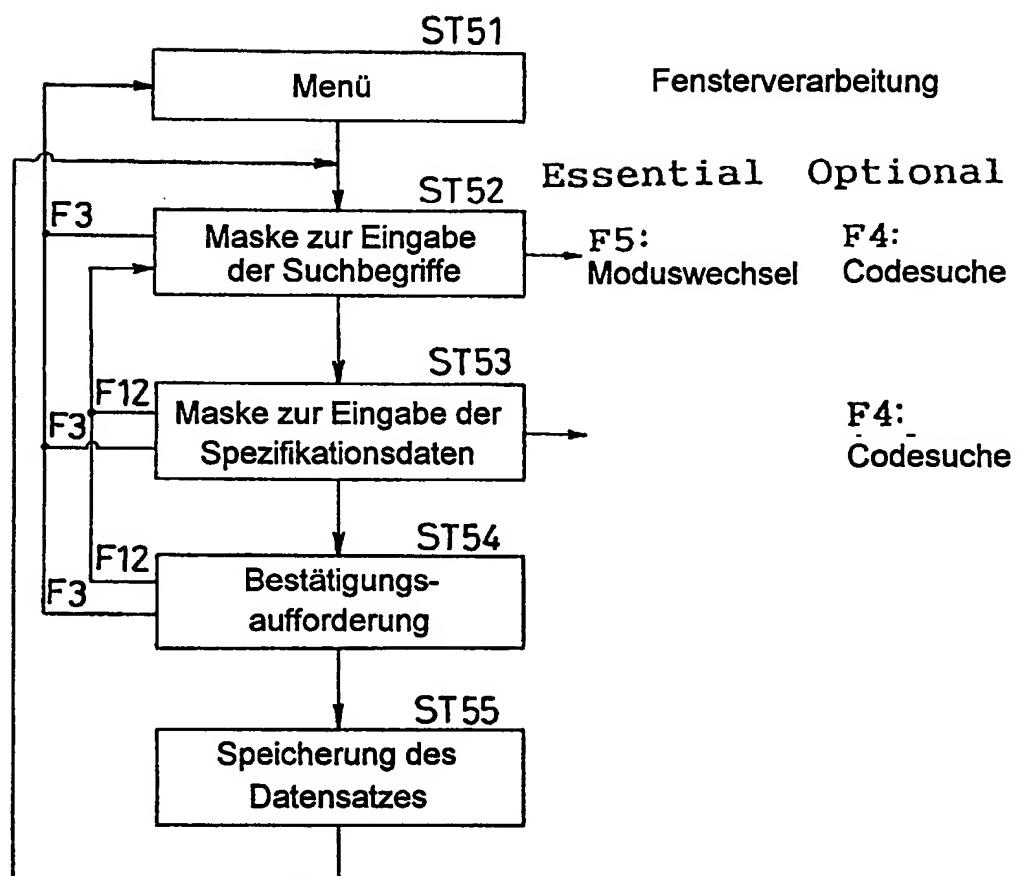


Fig.20

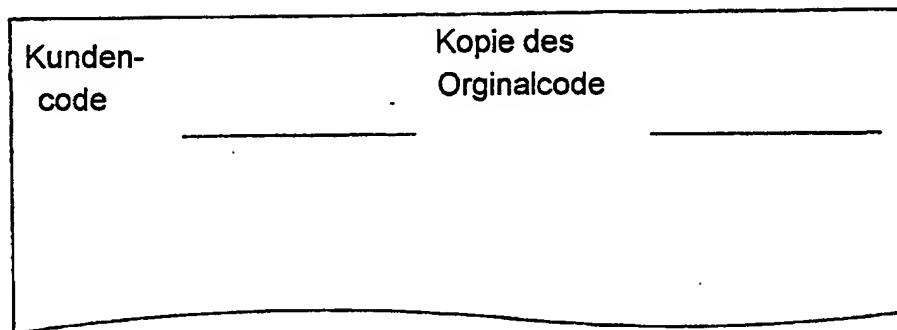


Fig.21

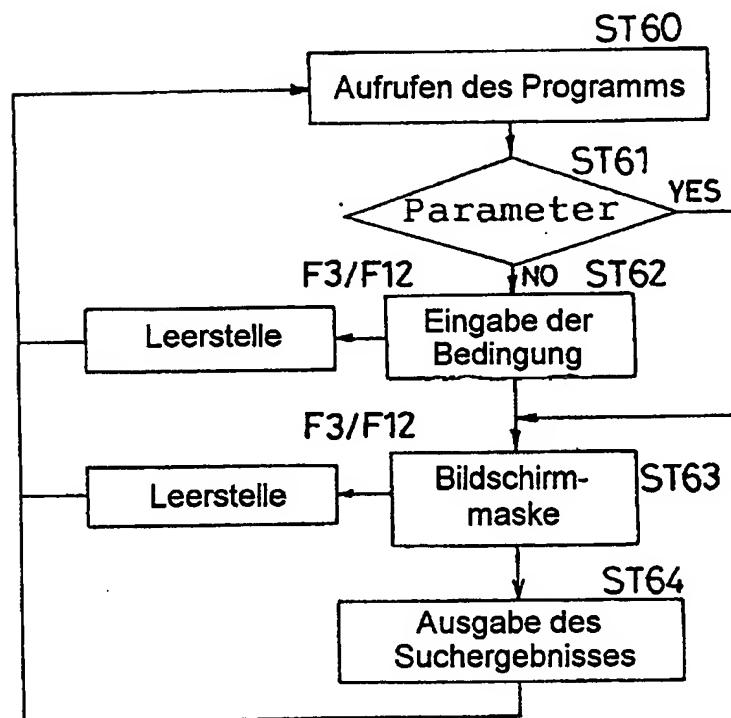


Fig. 22

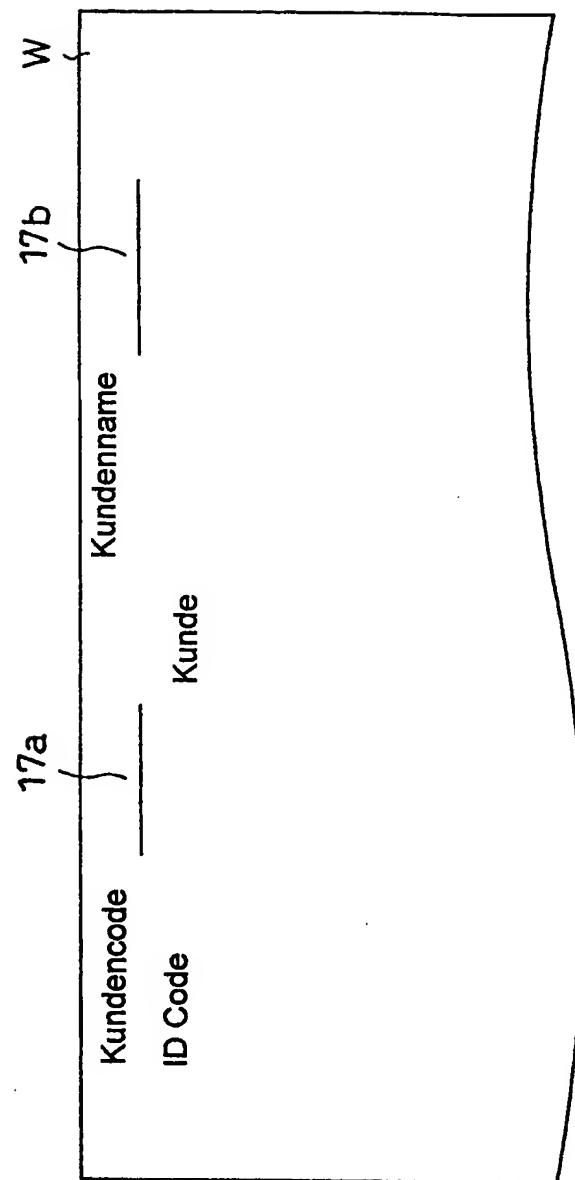


Fig.23

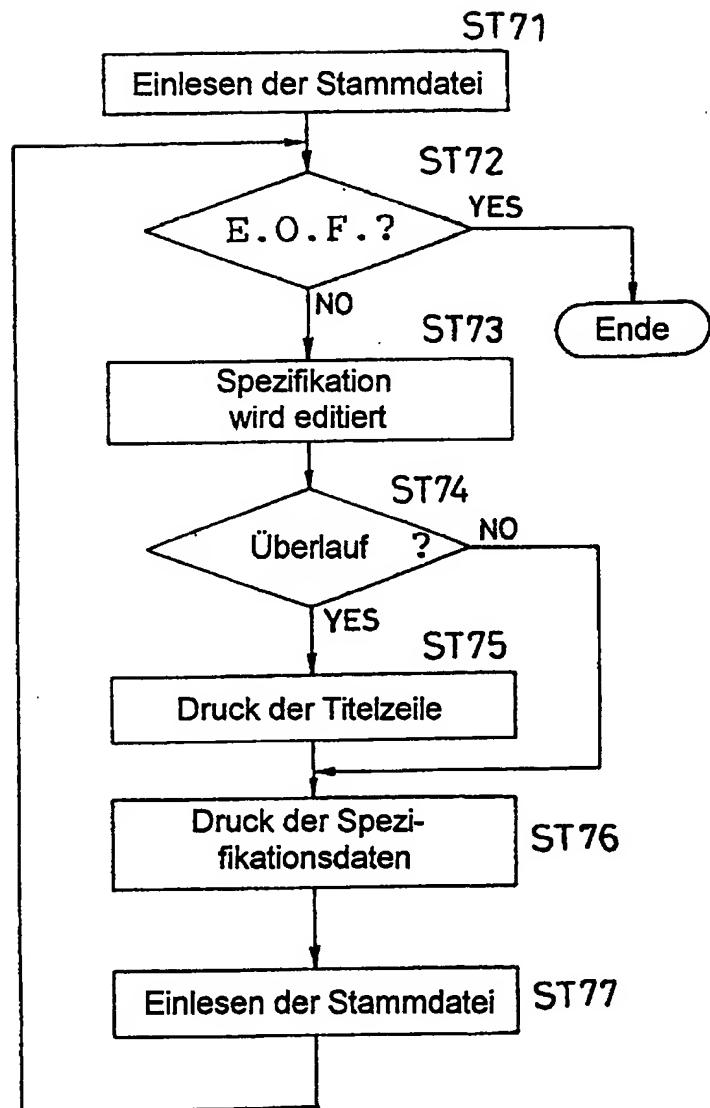


Fig.24

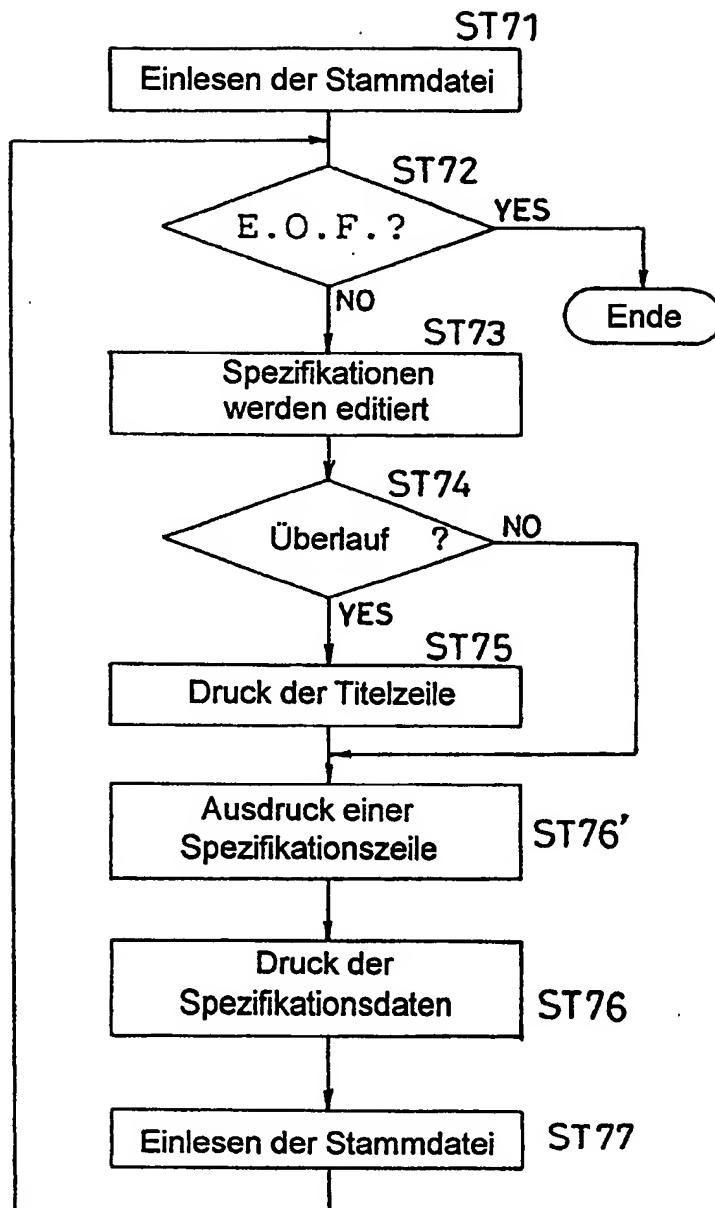


Fig. 25

HD→		Belegnr.	Datum	Oct. 3	Lieferant	ABC	Einkäufer	Sato
		Zeilennr.	Einkaufsgegenstand	Preis		Menge		
		1						
		2						
		3						

HD→		Belegnr.	Datum	Oct. 3	Lieferant	BTF	Einkäufer	Tanaka
		Zeilennr.						
		1						
		2						
		3						

Fig.26 Programmastein LT010

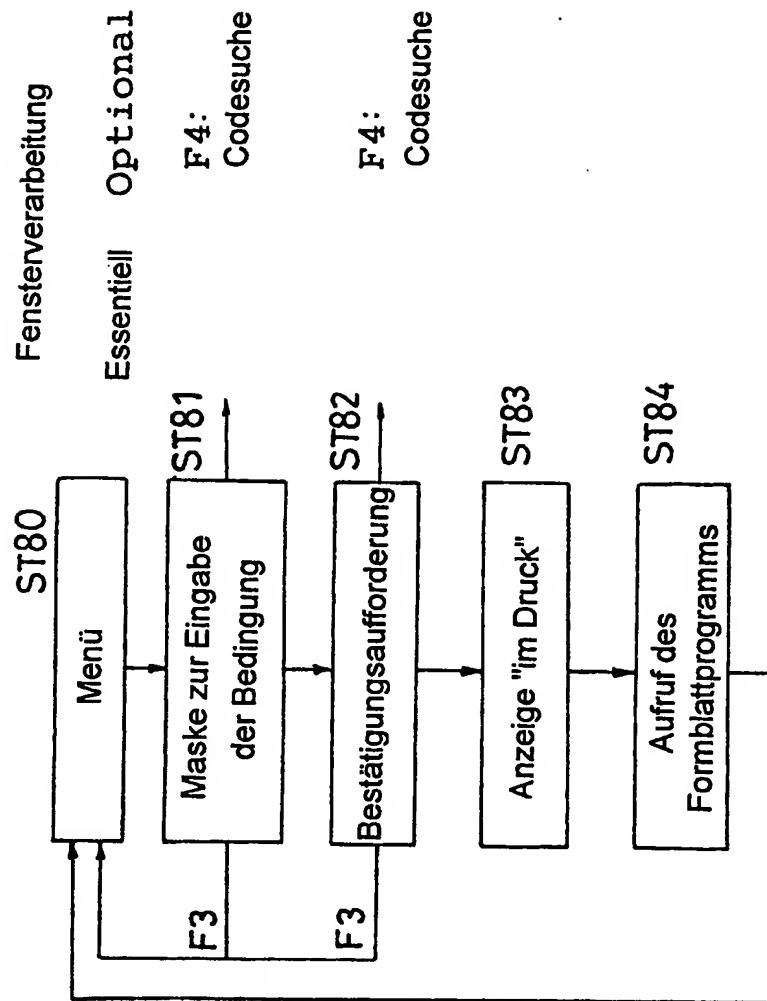


Fig.27

Programmbaustein LT030

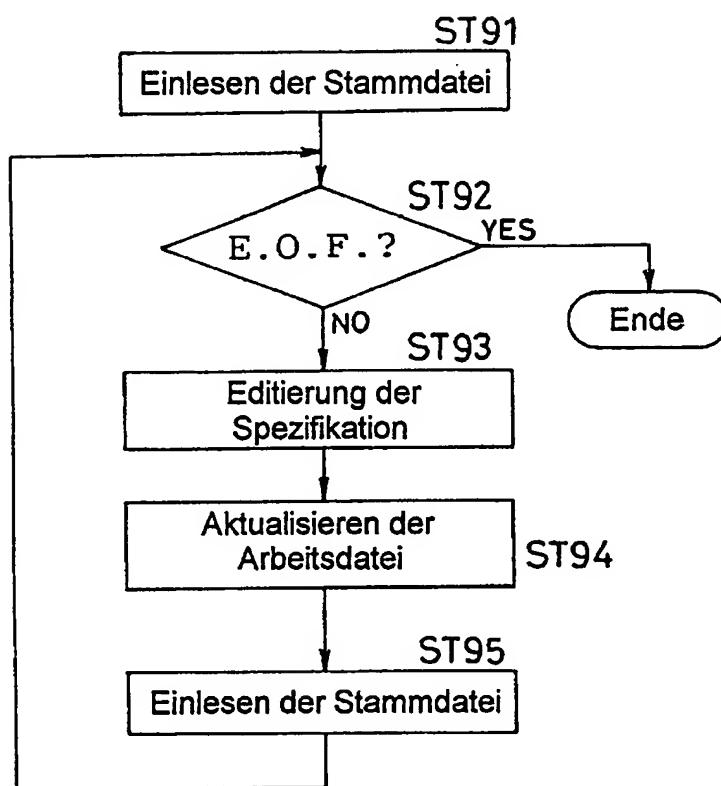


Fig. 28

Entwurfsmuster 33, 34
 Programmbaustein LT040

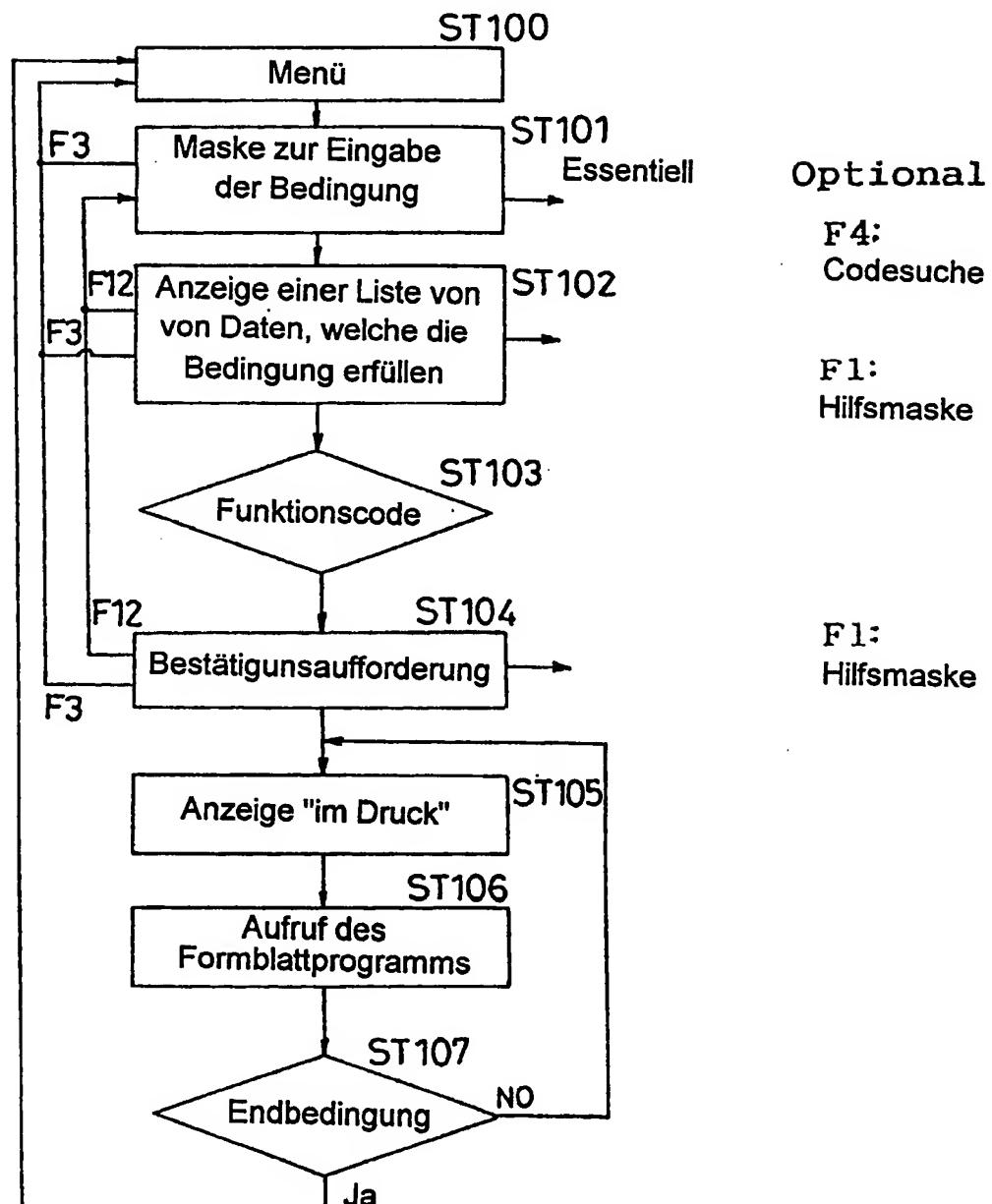


Fig.29

Entwurfsmuster 35(BU 020)

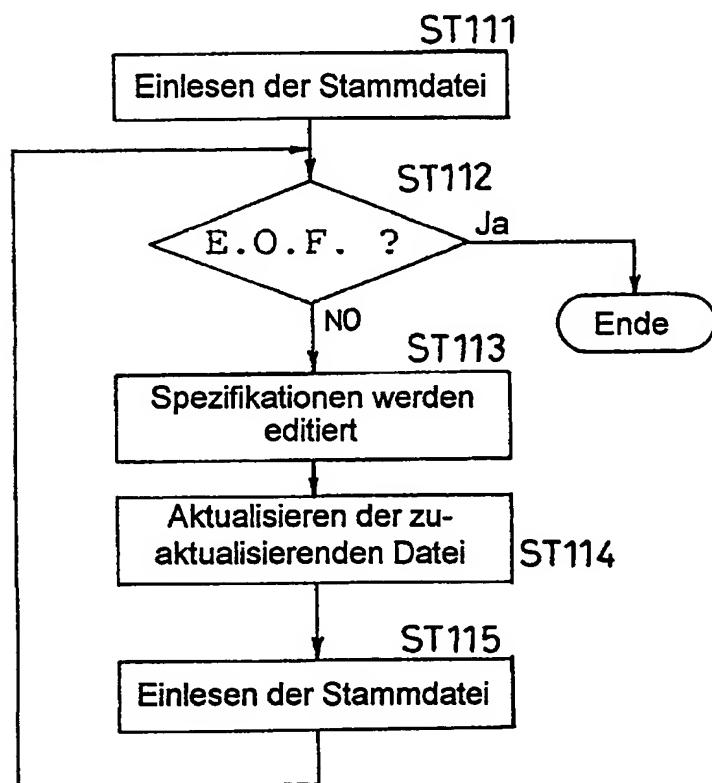


Fig.30

Entwurfsmuster 36(BU010+BU020)

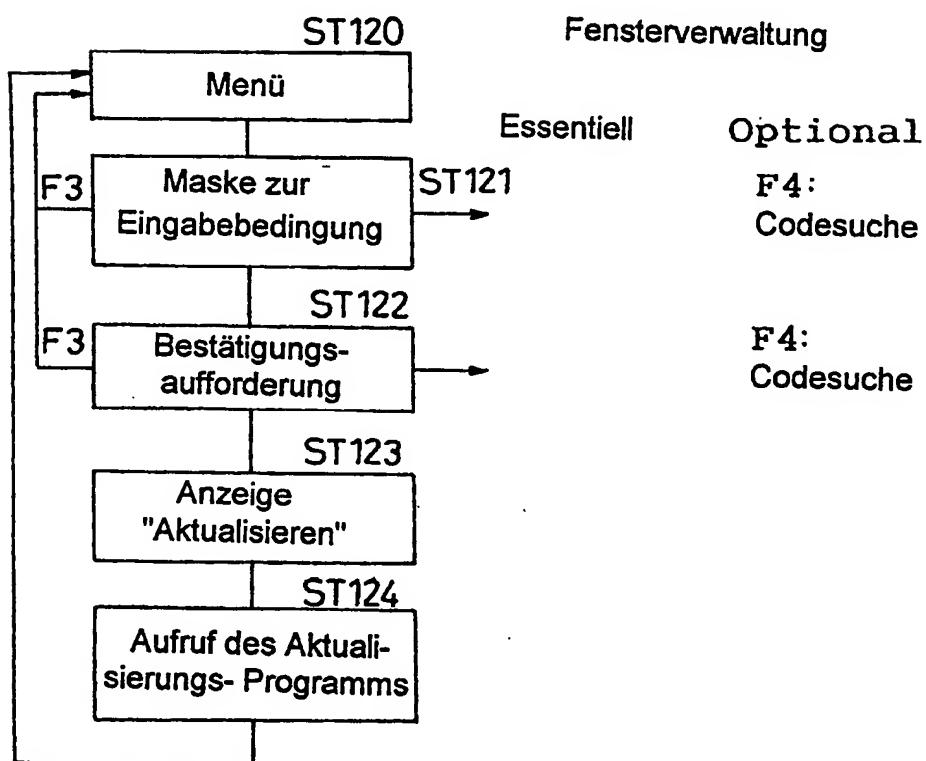


Fig.31

(Enwurf der physikalischen Datei)

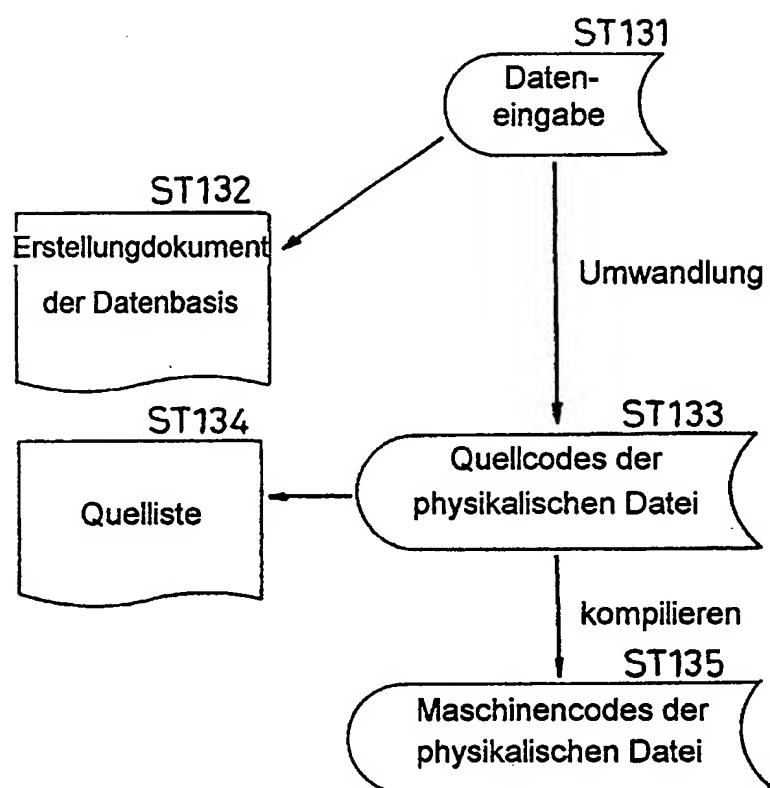


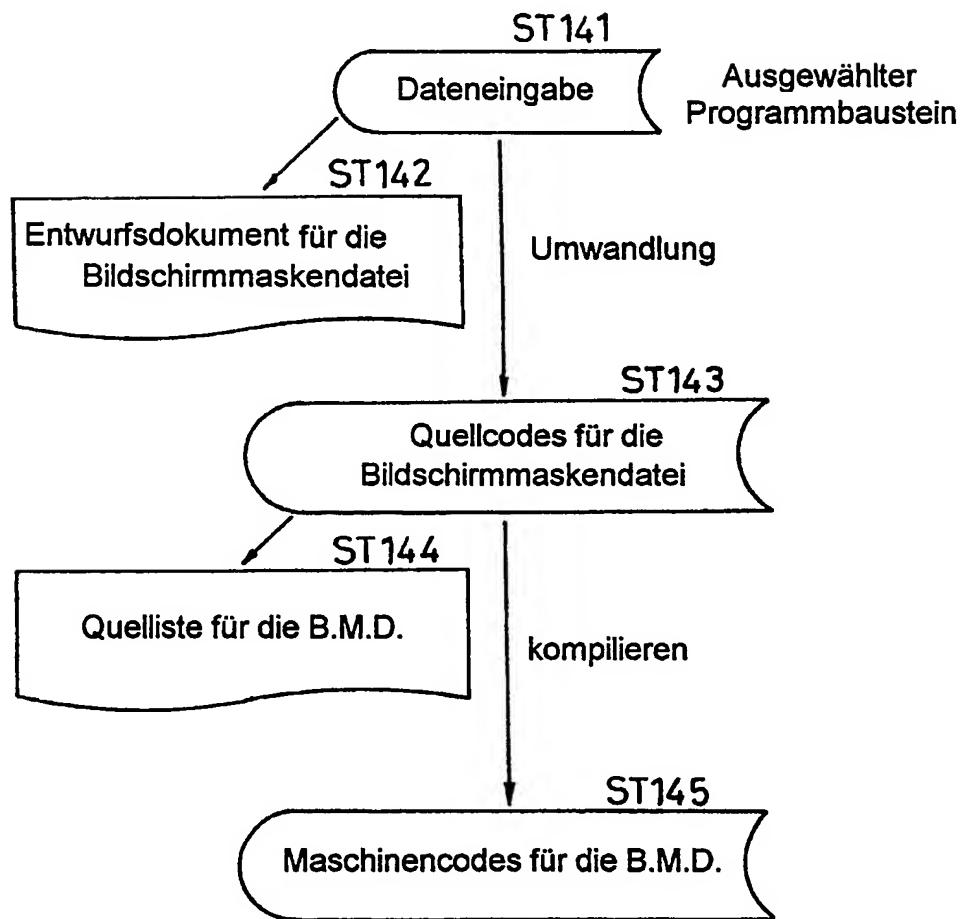
Fig. 32

20a 20b 20c 20d 20e

Fig. 33

Fig.34

Entwurf der Datei für
die Bildschirmmaske (B.M.D.)



३५

画面レイアウト	新規	...101	固定名	R/4000特申請
	画面 ID	R000001	画面名称	社員マスターメンテナанс

Fig. 36

画面レイアウト		用語名 画面 ID	別名 画面名	R./400件許用
		RC00001	画面名	社員マスターメンテナンス

Fig. 37

```

1.00 A=====
2.00 A*
3.00 A# 999101 COPYRIGHT RCC. 1994 LICENSED MATERIAL -
4.00 A# PROGRAM PROPERTY OF RCC.
5.00 A# REFER TO COPYRIGHT INSTRUCTIONS FROM
6.00 A# NO. RCC-999101
7.00 A#
8.00 A=====
9.00 A# -----
10.00 A# 画面ファイル名称 RC00D001 エカルト 本体No :MA011 *
11.00 A# 社員マスター・メンテナンス *
12.00 A# -----
13.00 A#<< キーワード定義 >>
14.00 A PRINT ~~
15.00 A HELP(G0 'ヘルプキー ')
16.00 A# -----
17.00 A# FMT01 : キ一部入力画面 *
18.00 A# -----
19.00 A R FMT01 TEXT(' キ一部入力画面 ')
20.00 A#<< キーワード定義 >>
21.00 A OVERLAY
22.00 A PROTECT
23.00 A CF03
24.00 A CF05
25.00 A#<< カーソル制御 >>
26.00 A CSRLOC(D1LIN D1COL)
27.00 A#<< カーソル制御用フィールド >>
28.00 A D1LIN 3 0II TEXT(' カーソル制御(行) ')
29.00 A D1COL 3 0II TEXT(' カーソル制御(列) ')
30.00 A#<< フィールド定義 >>
31.00 A DIPCID 10A 0 1 STEXT(' プログラムID ')
32.00 A DIGMNO 2A 0 1 1TEXT(' 画面No. ')
33.00 A DIMODM 6U 0 1 GOTEXT(' MODE ')
34.00 A DSPATR(RI HI)
35.00 A 1 71DATE EDTCODE(Y)
36.00 A 2 28' 社員マスター・メンテナンス '
37.00 A DSPATR(RI)
38.00 A 4 2' 社員コード '
39.00 A DS002 R A B 4 15TEXT(' 社員コード ')
40.00 A REFfld(EMPCODE RC1ENPP)
41.00 A DSPATR(UL CS)
42.00 A 30
43.00 A 4 24' 複写元コード '
44.00 A 61 DS001 R A B 4 39TEXT(' 社員コード(複写) ')
45.00 A REFfld(EMPCODE RC1ENPP)
46.00 A DSPATR(UL CS)
47.00 A 31
48.00 A 5 2' -----
49.00 A -----
50.00 A -----
51.00 A#<< 非表示フィールド定義 >>
52.00 A DS018 10A II TEXT(' バックアップ ')
53.00 A# -----
54.00 A# FMT02 : データ部入力画面 *
55.00 A# -----
56.00 A R FMT02 TEXT(' データ部入力画面 ')

```

Fig.38

Registrierung der Einheiteninformation

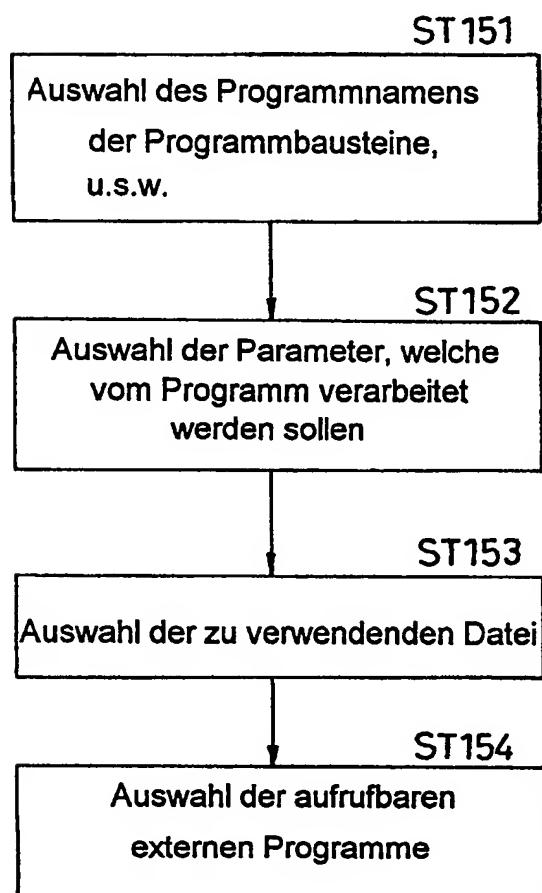


Fig.39

(Programmentwurf)

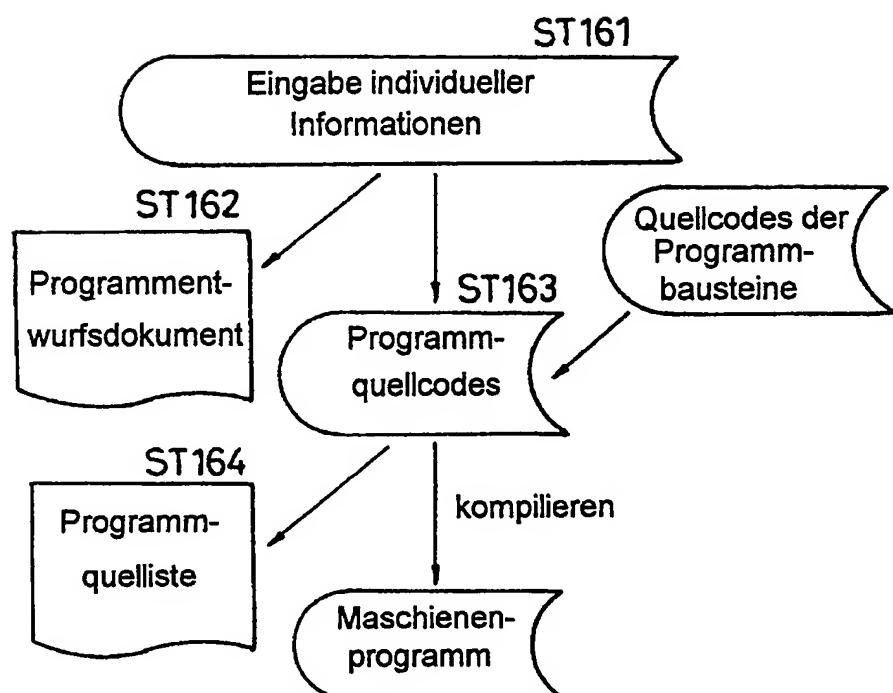


Fig.40

Fig.41

```

113.00// /*<< コントロールフラグ >>
114.00// WFLG      DS
115.00//   I          P 1 10WDSP
116.00//   I          2 2 WUDP
117.00//   I          3 3 WMD
118.00//   I          4 4 WMDT
119.00//   I          5 5 WCNL
120.00//   I          6 6 WCLR
121.00//   I          7 7 WIRR
122.00//   I          8 8 WIPC
123.00//   I          9 9 WADD
124.00// /*
125.00// /*
126.00// /* | ワークフィールドの定義
127.00// /* | W8DATE 8桁日付 (YYYYMMDD)
128.00// /*
129.00// /*
130.00// /*<< ワークフィールド >>
131.00// WKFLD     DS
132.00//   I 0          P 1 50W8DATE
133.00// C----- PARM LIST
134.00// C----- *
135.00// C----- *
136.00 C----- 社員マスターの検索 >>
137.00 C       PLIST1  PLIST
138.00 C       PARM      PECODE 6      社員コード
139.00// /*<< 日付チェックプログラム >>
140.00// C       PLSTD  PLIST
141.00// C       PARM      PeJCDY 80
142.00// C       PARM      PeJCTP 1
143.00// C       PARM      PeJCTT 1
144.00// C----- メッセージ送信 >>
145.00// C       PLMSG  PLIST
146.00// C       PARM      MSGID 7      MSG ID
147.00// C----- モード切替 >>
148.00// C       PLMOD  PLIST
149.00// C       PARM      PERIOD 1      モード
150.00 C----- *
151.00 C----- KEY LIST
152.00 C----- *
153.00 C----- 社員マスター >>
154.00 C       KEYEPH  KLIST
155.00 C       KFLD      EMPCODE      社員コード
156.00// C----- *
157.00// C----- MAIN ROUTINE
158.00// C----- *
159.00 C       MOVELOUROC  COROC      COPYRIGHT RCC
160.00// C----- 初期設定 >>
161.00// C       EXSR SBINZ      初期設定処理
162.00// C----- *
163.00// C       DO  $IIIVAL
164.00// C----- 画面処理 >>
165.00// C       WDSP      CASEQ0  SBEND      終了処理
166.00// C       WDSP      CASEQ1  SB1000     第1画面処理
167.00// C       WDSP      CASEQ2  SB2000     第2画面処理
168.00// C       ENDCS

```